



TECHNISCHE BERICHTE TECHNICAL REPORTS

ISSN 1430-211X

TUD-FI17-03-Dezember 2017

Prof. Dr. Frank J. Furrer, Georg Püschel (Eds.)
Institut für Software- und Multimediatechnik

From Algorithmic Computing to
Autonomic Computing

Hauptseminar im Sommersemester 2017

Prof. Dr. Frank J. Furrer

From Algorithmic Computing to Autonomic Computing

Editors: Prof. Dr. Frank J. Furrer, Georg Püschel

Technische Universität Dresden
Technical Report TUD-FI17-03-Dezember 2017



©Fotalia (with permission)

Table of Contents

Introduction	5
1 A Process Data Based Autonomic Optimization of Energy Efficiency in Manufacturing Processes <i>Daniel Höschele</i>	9
2 Eine autonome Optimierung der Stabilität von Produktionsprozessen auf Basis von Prozessdaten <i>Richard Horn</i>	25
3 Assuring Safety in Autonomous Systems <i>Christian Rose</i>	41
4 MAPE-K in der Praxis - Grundlage für eine mögliche automatische Ressourcenzuweisung in der Cloud <i>Michael Schneider</i>	59

Introduction

Prof. Dr. Frank J. Furrer

Context

In *algorithmic computing*, the program follows a predefined set of rules – the algorithm. The analyst/designer of the program analyzes the intended tasks of the program, defines the rules for its expected behaviour and programs the implementation. The creators of algorithmic software must therefore foresee, identify and implement all possible cases for its behaviour in the future application!

However, what if the problem is not fully defined? Or the environment is uncertain? What if situations are too complex to be predicted? Or the environment is changing dynamically? In many such cases algorithmic computing fails.

In such situations, the software needs an additional degree of freedom: ***Autonomy!*** Autonomy allows software to adapt to partially defined problems, to uncertain or dynamically changing environments and to situations that are too complex to be predicted. As more and more applications – such as autonomous cars and planes, adaptive power grid management, survivable networks, and many more – fall into this category, a gradual switch from algorithmic computing to autonomic computing takes place.

Autonomic computing has become an important software engineering discipline with a rich literature, an active research community, and a growing number of applications.

History

Autonomic computing ([1], [2], [3]) has been introduced by IBM in 2001: The reason was the growth in complexity of modern computer installations, which was becoming unmanageable. The basic idea was to develop management software with *self-* properties*. The originally proposed set of properties were *self-configuration, self-healing, self-optimization and self-protection*. Autonomic computing systems therefore are able to re-organize, optimize, defend and adapt themselves with no real-time human intervention ([6]). Autonomic computing relies on many branches of science – especially computer science, artificial intelligence, control theory, machine learning, multi-agent systems and more.

Autonomic Computing Architecture

The key element proposed by IBM was the *autonomic computing architecture*: The MAPE-K architecture (= **M**onitor – **A**nalys(e) – **P**lan – **E**xecute – **K**nowledge). This architecture is basically a modern control loop, reading state values from the physical computer installation, processing it through the MAPE-K loop and acting on the physical computer installation ([3], [6], [7]). The interesting part is the “K” – for knowledge: The MAPE-K loop utilizes technologies from the field of artificial intelligence ([8], [9]) in order to reach various degrees of autonomy.

Cyber-Physical Systems

Soon after the introduction of the MAPE-K architecture, it became widely used in the construction of **cyber-physical systems** ([6], [7]). Cyber-physical systems (CPS) are cooperating networks of physical and *computational* components. In a CPS – such as a modern car, plane or train – the physical components are *controlled by software*. The “intelligence” is therefore implemented in programs with varying degrees of autonomy, often using technologies from the field of artificial intelligence.

Seminar Work

This seminar will work on the central theme:

Which are the history, the technologies and the applications of Autonomic Computing?

Each participant chose one of the 3 fields:

- F1** Research and describe the *historical development* from algorithmic computing to today’s autonomic computing;
- F2** Investigate which *autonomic computing technologies* fill the MAPE-K architecture today and in the future;
- F3** Identify and document a number of *promising applications* of cyber-physical systems (CPS) based on autonomic computing.

The Hauptseminar has *3 seminar days* (see separate work program, dates below):

- An introduction day: Autonomic Computing will be introduced in a lecture by Professor Dr. Frank J. Furrer, and the parts of the Hauptseminar (Paper, presentation) will be defined,
- Individual, guided research in the selected area and authoring of a scientific paper. Feedback from peer reviewers,
- A first seminar day: The participants will present their results and receive feedback from the audience,
- Improvement of the paper and the presentation, based on the peer feedback,
- A second seminar day: The participants will present their improved results and receive feedback from the audience,
- Delivery of the final paper.

The participants will learn: (a) to do focused research in a specific area (“Autonomic Computing”), (b) to author a scientific paper, (c) to improve their LaTeX expertise, (d) to experience the peer-review process and (e) to hold convincing presentations, and (f) to benefit from a considerable broadening of their perspective in the field of technology, software, and applications.

As a final outcome of the seminar, a *proceedings* volume – including all the papers produced by the participants – will be assembled and made available in electronic form to anybody interested.

Seminar language is English. Three seminar days will be held and 3 ECTS credits are awarded for the successful participation. Audience is limited to 7 participants. Please register in advance.

Seminar Schedule

<u>Kick-Off Meeting (Introduction):</u>	Wednesday, April 19, 2017 / 11:10 – 12:40 in APB/INF 2101
<u>Seminar Day 1:</u>	Wednesday, June 7, 2017 / 09:20 – 10:50 & 11:10 – 12:40 in APB/INF 2101
<u>Seminar Day 2:</u>	Wednesday, July 12, 2017 / 09:20 – 10:50 & 11:10 – 12:40 in APB/INF 2101

References

1) Mandatory Reading

- [1] The fundamental knowledge:
Philippe Lalanda, Julie A. McCann, Ada Diaconescu: **Autonomic Computing – Principles, Design and Implementation**. Springer-Verlag, London UK, 2014. ISBN 978-1-4471-5006-0
- [2] The seminal work:
IBM Research Paper, 2001: **Autonomic Computing – IBM’s Perspective on the State of Information Technology**. Downloadable from: http://people.scs.carleton.ca/~soma/biosec/readings/autonomic_computing.pdf [last accessed: 2.2.2016]

- [3] Introduction to the Architecture:
IBM White Paper: *An architectural blueprint for autonomic computing*. 3rd edition, June 2005. Downloadable from: <http://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf> [last accessed: 2.2.2016]
- [4] Cyber-Physical Systems:
NIST Engineering Laboratory: *Cyber-Physical Systems*. Downloadable from: <https://www.nist.gov/el/cyber-physical-systems> [last accessed: 2.12.2017]

2) References

- [5] Mark Klein, Raj Raikumar, Dionisio De Niz: **Cyber-Physical Systems**. SEI Series in Software Engineering. Addison Wesley, USA, 2016. ISBN 978-0-321-92696-8
- [6] Manish Parashar, Salim Hariri: **Autonomic Computing - An Overview**. In: J.-P. Banatre et al. (Eds.), UPP 2004, LNCS 3566, pp. 247–259. Springer-Verlag Berlin Heidelberg, 2005. Downloadable from: <http://143.106.148.168:9080/Cursos/IA844/02-11/autonomic.pdf> [last accessed: 23.1.2017]
- [7] Payal Mittal, Abhay Bansal, Abhishek Singhali: **A Study on Architecture of Autonomic Computing Self Managed Systems**. International Journal of Computer Applications (0975 – 8887), Volume 92, No.6, April 2014. Downloadable from: <http://research.ijcaonline.org/volume92/number6/pxc3894890.pdf> [last accessed: 23.1.2017]
- [8] **Artificial Intelligence – Overview**. Downloadable from: https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_overview.htm [last accessed: 23.1.2017]
- [9] Stuart J. Russell, Peter Norvig: **Artificial Intelligence – A Modern Approach**. Prentice Hall International, USA, 3rd revised edition, 2016. ISBN 978-1-292-15396-4

A Process Data Based Autonomic Optimization of Energy Efficiency in Manufacturing Processes

Daniel Höschele

TU Dresden

Abstract. Energy efficiency is not only a key factor for the global economy but also for the industrial economy. The last few years have reported a rise of energy costs, which increase the costs of companies and infect their competitiveness. The present paper aims to optimize the energy efficiency of industrial processes, that are based on Cyber Physical Systems in the context of Autonomic Computing. Therefore, an architecture will be provided to optimize the energy efficiency of the whole process. In addition, it will be illustrated how the energy consumption of an industrial process can be predicted with the Naïve Bayes classification and how the process can be regulated within the framework of the provided architecture.

Keywords: Self-organizing autonomic computing · Cyber-physical networks · Industry and manufacturing · Process control systems · Process Mining · Energy efficiency

1 Introduction

A positive energy supply, low emission of carbon as well as low energy costs and the dependency from resource rich countries has increased the interest in energy relevant topics in companies. Germany for example passed in the year 2000 the Renewable Energy Law to support power plants, which produce renewable energy. Thus, these costs are allocated to the end-consumer. Therefore, not only the power costs are increasing for companies but also their competitiveness is decreasing [1]. Some authors claim that energy management, which is supported by modern information and communication technology is indispensable in the future [2]. Especially the manufacturing sector is affected by those changes.

The following paper aims to illustrate an architecture how a process chain of Cyber Physical Systems (CPS) can adapt itself to reach the best possible energy efficiency based on the concepts of autonomic computing. The first part of the paper deals with fundamental terms and frameworks like Autonomic Computing, process data or energy efficiency. Afterwards an architecture will be introduced, which is based on the basic Autonomic Computing architecture by IBM. The last part of the paper deals how energy efficiency can be predicted based on process data.

2 Fundamentals

2.1 Autonomic Computing

In October 2001 IBM claimed that maintaining Software is a main obstacle for further progress because of the complexity and dependencies caused by million lines of code [3]. Modern computing systems challenging the human capability to deal with software complexity in a new way. Autonomic Computing (AC) aims to minimize human interventions and to create self-regulating computer systems with the prime goal to deal with predictable as well as unpredictable changes [4]. In addition, AC aims to reduce the time and effort for administration as well as costs, which are produced by extensive administration activities nowadays [4]. Furthermore, the goals of AC highly depending on high-level policies or objectives of the company. To characterize Autonomic Systems (AS) there are four key properties, which describe the key features of such a system. As mentioned before AS are systems with a **self-management** capability. This means that an AS will adjust and maintain it's operations in course of external and internal changes. Changes for example are varying components, workloads, demands as well as software or hardware failures [3]. In addition, an Autonomic System is capable to configure itself, which is called **self-configuration**. The system will accord itself according to the high-level policies as well as objectives. Supplementary those systems are able to configure new elements without external intervention in the network of autonomic systems. Like a new person in a social network, the autonomic systems would be capable to embed the person in the social network and would know about the person friends, links, etc. In addition, an autonomic system is capable to optimize and heal itself. That means these systems can tune to their parameters in the best possible way and are able to detect, diagnose and repair localized problems [3]. Besides the four key properties there are a few number of extended capabilities, which come along autonomic systems like the prediction of future events (*self-anticipating*) and the ability modify itself during runtime.

Architecture of Autonomic Systems Besides the term autonomic computing IBM also suggested a layer architecture for autonomous systems, which is illustrated below. The lowest layer consists of all hardware and software components, which make up the basic IT infrastructure. Above the first layer there is a touchpoint layer, which acts as interface for the hardware below. Layers three and four automate some portion of the IT process using an Autonomic Manager (AM) [5]. Layer four contains autonomic managers, that orchestrate the managers of layer three. The last layer provides a common system for the IT professional to interact with the system [5]. The illustrated architecture provides the basic framework for the introduced architecture in chapter 3.

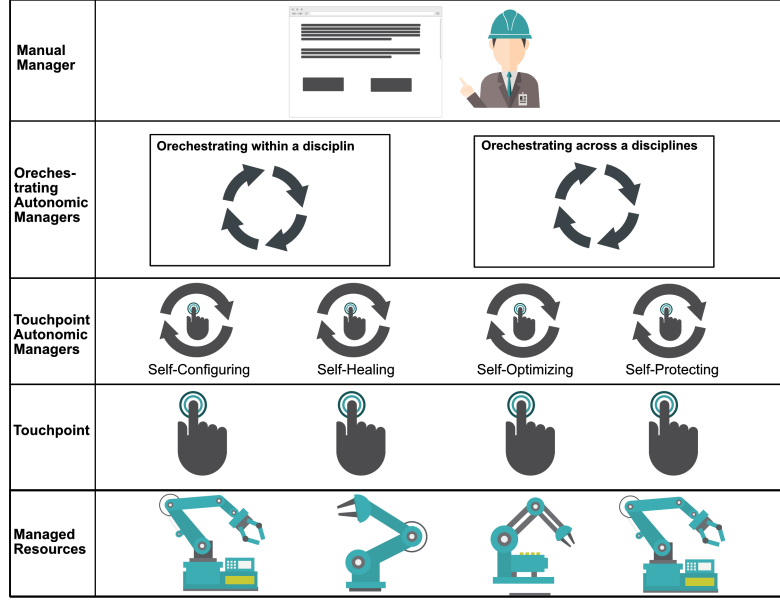


Fig. 1. The autonomic architecture

2.2 Cyber-Physical Systems

The miniaturization of semiconductors and new demands in terms of quality and lower costs determine the semiconductor industry nowadays. At the same time the classic production is changing to a more fully automated and smarter production [5]. Cyber Physical Systems are the fundamental of many future visions and scenarios [6]. The usage of CPS shows a great promise to optimize the production process of an organization, especially in the context of autonomic production where human intervention will be minimized [6].

In literature, there are a lot attempts to characterize CPS. In the following some definitions are listed.

Table 1. Definitions of Cyber Physical Systems

Lee (2013)	<i>"Cyber-Physical Systems (CPS) are integrations of computation and physical processes. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa."</i> [7]
Kaestle et al. (2013)	<i>"CPS are characterized as self-describing systems with own intelligence, which dispose of an autonomous decentralized processing unit and are able to communicate directly over the internet."</i> [8]
NIST (2016)	<i>"Cyber-physical systems (CPS) are smart systems that include engineered interacting networks of physical and computational components."</i> [9]

In [7], the author characterizes Cyber Physical Systems as an integration of computation and physical processes [7]. In addition, "CPS are able to communicate, to perceive the environment, to interpret information and to act on the physical world". The third Definition connect to the upper definitions and aims that CPS are divided into physical and computational components. The National Institute of Standardization and Technology (NIST) added one important fact, that CPS are not a single component but that a Cyber Physical System is an interacting network of different components. According to those definitions, CPS commemorate on Embedded Systems ¹.

The NIST aimed that there are a lot of domains, which can profit from CPS Systems. Especially efficiency gains in the manufacturing environment are expectable. The improvement of energy efficiency is one part of that. Based on the increasing potential of CPS, which are enabled by new sensor technology, the frequency of collecting energy data also has increased [9]. The produced amount of energy data will open new doors in optimizing the energy efficiency of processes, which are based on CPS.

In the context of the present paper a Cyber Physical System will be understood as a computational unit with own intelligence, based on computational components. Additional a CPS is able to communicate with it's environment and interact with physical parts.

¹ "Embedded systems are information processing devices which form often miniaturized components of larger computer systems. Every component has a specific functional purpose. In combination with each other, they determine the value proposition of the entire system" [6].

2.3 Process Data

In general, a process consists of series of events to create a specific output. Furthermore, processes consists of specific start-events and require different inputs [9]. Other authors claim that process activities can be carried out parallel from different organization units and that processes are measurable [10]. In the context of this paper a process consists of a set of activities, which are measurable. The process has always one start event and one end event, which can be triggered by a sequence of activities.

The discipline, which deals to mine information from processes is called Process Mining. Process Mining is a relatively young research discipline, which aims to extract valuable process-related information from Event-Logs [11]. Process Mining also present a data structure for processes and a very good definition for process variants in this context. In the following it will be explained how process data should look like and what a process variant is. The discipline of Process Mining will not be explained because of the complexity which comes along with it.

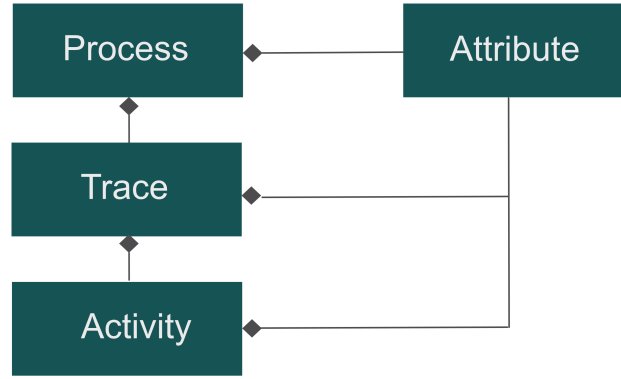


Fig. 2. Structure of process data

In general, the process is discovered from different Event-Logs. Such logs are containing all process relevant data. As illustrated in Fig. 2 a process consists of a set of cases, which contains different activities. A case stands for a process instance, which symbolize a unique process run. An activity stands for an action of the process. It consists of different attributes like name, time, date, and other optional attributes. This implicates that process data can consist of categorical and continuous attributes. Besides the basic process terms the discipline of Process Mining also introduced the term of process variants in a new way. A process variant is a unique sequence of events from the start till the end of the process. An example is illustrated in Fig. 3 below.

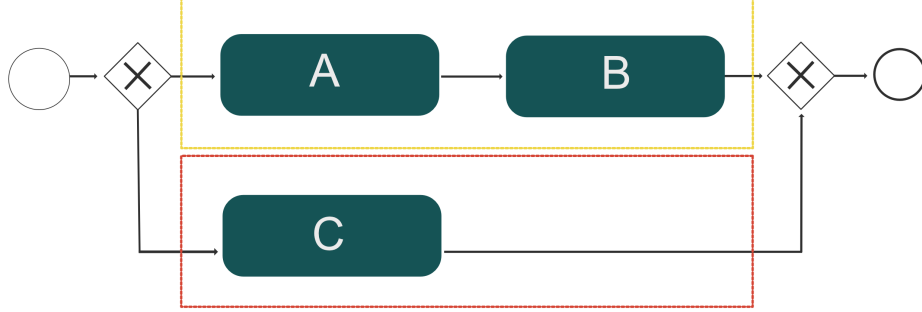


Fig. 3. Process Variants

The example above illustrates two process variants: $\langle A, B \rangle$ and $\langle C \rangle$. These two options are the unique sequences from the start of the process to its end. Such process can be described in a much more understandable way like in appendix A, which serves as an example for the further work.

To perform an energetic optimization, it is necessary to store the energy consumption in the event log as an own attribute [12]. If we take a closer look on the manufacturing processes based on CPS, each Cyber Physical System represents a single activity in the process. Therefore, each CPS should be aware of his energy consumption and connected to the manufacturing process. Furthermore, the process discovery of event logs or process data allows to analyze processes respective to their variants [12].

2.4 Energy Efficiency

Energy Efficiency is a relatively generic term. There is no consistent measure for it. The fundamental understanding of energy efficiency is to require less energy to produce the same amount of one product. Energy in general is the ability to perform work [13]. There are a lot of attempts in literature to define the term "energy efficiency". In the context of the present paper the following definition will be used:

"In general, energy efficiency refers to using less energy to produce the same amount of services or useful output. [...] Hence, energy efficiency is often broadly defined by the simple ratio:

$$\frac{\text{Useful output of a process}}{\text{Energy input into a process}} \quad (1)$$

"[14].

Patterson describes energy efficiency as a ratio between useful output of a process and the corresponding input. Within the context of the present paper energy efficiency will be defined as the ratio of the consumed energy of the

processes and the number of process instances. The energy consumption is the sum of all consumptions of each activity in the process.

The crucial factor to minimize energy efficiency is the energy input, which is needed by the process to produce a specific amount of some product. Thus, energy efficiency means to reduce the use of energy in a system, which is used to produce a product. On closer examination, the definition does not specify what the output is. Therefore, the produced output is not in the scope of the present paper. The paper will focus how to autonomically minimize the energy consumption in a process, independent from the produced output. Thus, a lower energy consumption also implicates a higher energy efficiency of the process.

3 A Potential Architecture

As mentioned above the architecture of an autonomic system consist of different components. Normally it is a five-layer architecture, which describes the fundamental structure of an autonomic system. The outlined architecture illustrates how a process chain of CPS can adapt their energy efficiency in an autonomic and automatic way. To demonstrate the main idea of the architecture different layers are summarized. Fig. 4 illustrates the architecture.

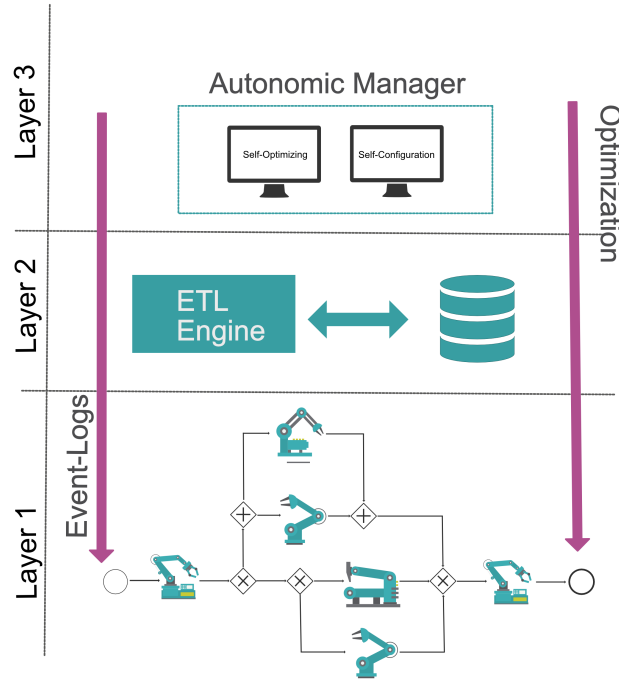


Fig. 4. An Architecture for an autonomic optimization of energy efficiency

The fundament of the given architecture is a process which consists of CPS. That means that each CPS operate in a sequence to produce a specific output. Thus, the first layer represents a Cyber-Physical System of Systems, which is characterized as a distributed network consisting of CPS [4].

Using the example of manufacturing processes the output can be all kinds of outputs for example a technical device like a phone or a mechanical device like a car. For that reason, it is useful to calculate the energy efficiency with the number of process instances, to provide a general framework. As mentioned above each of the single CPS produces event logs in which process relevant information and the energy consumption of the device are stored. The Layer above is responsible to parse and transform each event log into a single data source. This layer enables a holistic process-view and is the fundament for further analysis. The last layer is based on the data, which are produced by the second layer. With the knowledge of the whole processes it first discovers the event log and tries to optimize the process according to its energy consumption.

According to the explained features of Autonomic Computing the feature of self-management is reached by the database, in which the process data are stored. If the process changes, new data will be added to the historical data and the autonomic manager can react on those changes. Therefore, the architecture is able to configure itself, by predicting and calculating the energy consumption at every new starting process instance. Thus, the process can be regulated autonomically at every new starting process instance by choosing the process sequence with the lowest energy consumption.

3.1 Layer 1 & Layer 2 – Extracting and Transformation of Process Data

The idea is to analyze event data from a process-oriented perspective. It allows us to answer questions like [15]:

- What really happened in the past?
- What is likely to happen in the future?
- How to control a process better?

This are just a few questions how process mining can help us to understand data intense processes. But there are also a few issues, which come along extracting event logs. The starting points for extracting event-data are normally all kinds of data sources. The reality show that event data is often hidden in different data sources [12]. However, there are a few assumptions we will assume to guarantee a stable architecture:

1. All necessary data will come from Cyber-Physical-Systems in the process chain
2. Each CPS is one single data source, which is used to extract the relevant process information
3. All CPS can document their energy consumption

4. The energy consumption from one CPS differs from another CPS
5. The process only consists of Cyber Physical Systems
6. Every Process sequence leads to a unique end event.
7. The only attributes, which are needed are the time-stamp for each activity, the activity name, the energy consumption and an identifier for the process instance

The first two points are intended to ensure, that CPS can transmit all relevant process information to the ETL-Engine. Point three and four are describing on the one hand that CPS must be able to know, transmit and store their energy consumption. On the other hand, point four illustrates an important fact – it has to matter which process activity is carried out because each activity can have a different energy consumption. For example, in industrial processes it is possible that there are different tools from different manufactures, which have the same objective but have different energy consumptions. Another scenario can be that the energy consumption depends on the activities, which were carried out before.

But even with those assumptions there are still several problems, which can occur. Different data sources may use different formatting conventions. For example, to identify an activity one CPS uses a special key another CPS stores the activity name in English. Or one data source may use the date format "18-01-1993" and another CPS uses "1993/18/01". Also, the energy consumption can be stored in different measures, which is bad for further analysis. In the context of the business intelligence the phrase "*Extract, Transform and Load*" is used to describe a process that involves extracting data from different sources, dealing with syntactical and semantical issues and loading the transformed data into a target system [15]. The following process illustrates how the single event logs can be transformed into a target system.

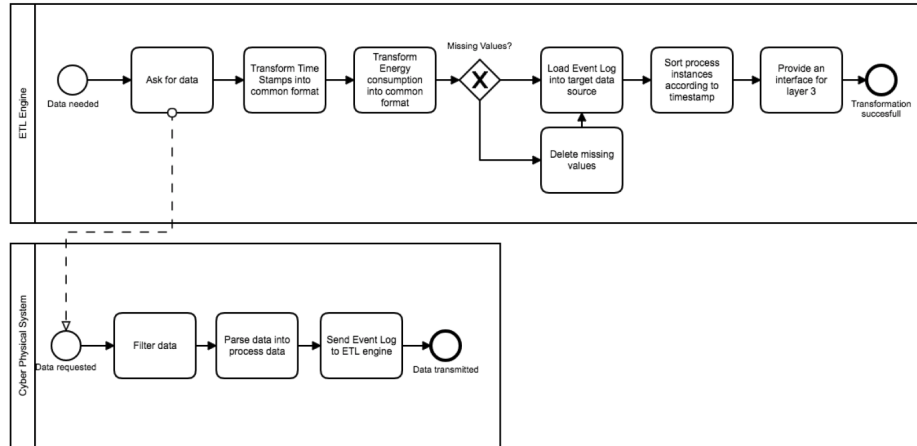


Fig. 5. The ETL Process of Layer 2

As illustrated above, the process consists of two participants. First the CPS and the ETL Engine, which is responsible to transform the data and provide it for the layer above. The main task of the CPS is to filter all necessary information and to parse them into some process mining related format, which is explained in chapter 2.3 and send them back to the ETL Engine. The first task after receiving the data is to convert all time stamps and energy metrics into a common format. In the case of missing values, such processes instances should be deleted in the case of process mining [12]. Afterwards the event-log is loaded into a target system in which other logs are stored.

The described process is very important to guarantee a consistent data format on which layer three can perform. Without the described process, it is very likely that the autonomic manager will have much more problems to deal with process data.

3.2 Layer 3 – The Optimization of the Event Log

Layer three is responsible for the energetic optimization of the process. All relevant process information where given by the layers one and two. Layer three has basically two main functions: first to **predict the energy consumption** of the process and second to **regulate the whole process**. At first, it will be explained how to predict the energy consumption based on data mining techniques. Based on the prediction of the energy consumption the autonomic manager will decide, which sequence in the process will follow.

Before we illustrate the method of Naïve Bayes classification to predict the energy consumption, tab. 5 must be transformed. The following table is illustrating the transformed table from appendix A. For each activity in the process we will introduce a dummy variable, which can be 1 or 0 to indicate the absence or presence of an activity in the process. To demonstrate the Naïve Bayes classification, we will use categorical attributes instead of continuous variables. For that reason, we need to classify the energy consumption in different groups. Values from 0-150 belong to the class *low*, values from 150-200 belong to the class *middle* and values over 200 to the class *high*. At this point we transformed the nominal attribute activity to a binary attribute with the help of dummy variables and we transformed the energy consumption from a continuous attribute to a categorical attribute.

Table 2. Transformed process data

	A	B	C	D	E	F	Energy Consumption	Class
1	1	1	1	1	1	1	0	185 middle
2	0	1	1	1	1	1	0	135 low
3	0	1	1	1	0	1	1	205 high
4	1	1	1	1	1	1	0	190 middle
5	0	1	0	1	1	1	0	85 low
6	1	1	1	1	1	1	0	205 high

In comparison to other data mining procedures for classification problems like decisions trees, the Naïve Bayes classification tries to calculate the most likely classification. "The Naïve Bayes algorithm gives us a way of combining the prior probability and conditional probabilities in a single formula, which we can use to calculate the probability of each of the possible classifications in turn. Having done this we choose the classification with the largest value" [16]. That means, the method tries to predict the probability of the energy consumption for every process instance by calculation the conditional probabilities of every activity with the prior probability of the class and chose the classification with the highest value. With the use of process data, we need to introduce one restriction. As mentioned before we use dummy variables to symbolize if an activity occurs in a process sequence or not. Activities which not occur in a given process instance shouldn't be considered for two reasons: first, the probability of an event which does not occur in the process sequence doesn't matter and second, the conditional probability would always be 0 for this activity.

The prior probability of an event will be indicated by a capital letter P. According to tab. 3 the prior probabilities for the 3 classes is $2/6$. The value represents how often the class occurs in the data set. "The probability of an event occurring if we know that an attribute has a particular value [...] is called the conditional probability of the event occurring" [16]. It is also written with a capital letter P and a condition. For example the probability that the activity A occurs in a process sequence given that the energy consumption is middle is written with the term $P(A = 1 \mid \text{Class} = \text{"middle"})$. To calculate this probability, we count the number of $A = 1$ where the class is equal to middle and divide them by the number of classes, which are middle. So the prior probability $P(A = 1 \mid \text{Class} = \text{"middle"})$ is equal to $1/2$ ($2/2 = 1$). Tab. 3 illustrates all conditional and prior probabilities of the example data set.

Table 3. Conditional and prior probabilities of Appendix A

	Class = Low	Class = middle	Class = High
A = 1	$0/2 = 0$	$2/2 = 1$	$1/2 = 0.5$
A = 0	$2/2 = 1$	$0/2 = 0$	$1/2 = 0.5$
B = 1	$2/2 = 1$	$2/2 = 1$	$0/2 = 0$
B = 0	$0/2 = 0$	$0/2 = 0$	$0/2 = 0$
C = 1	$1/2 = 0.5$	$2/2 = 1$	$2/2 = 1$
C = 0	$1/2 = 0.5$	$0/2 = 0$	$0/2 = 0$
D = 1	$2/2 = 1$	$2/2 = 1$	$1/2 = 0.5$
D = 0	$0/2 = 0$	$0/2 = 0$	$1/2 = 0.5$
E = 1	$2/2 = 1$	$2/2 = 1$	$2/2 = 1$
E = 0	$2/2 = 1$	$0/2 = 0$	$0/2 = 0$
F = 1	$0/2 = 0$	$2/2 = 1$	$1/2 = 0.5$
F = 0	$2/2 = 1$	$0/2 = 0$	$1/2 = 0.5$
Prior Probability	1/3	1/3	1/3

To determine the energy consumption of a process instance we calculate the conditional probabilities with the prior probability of the energy consumption for each class and choose the classification with the largest value. For example if we want to predict the energy consumption for the sequence $\langle A, B, C, D, E \rangle$ the calculation is:

$$\begin{aligned} \underline{Class = low} \qquad \qquad \qquad & 1/3 * 0 * 1 * 0.5 * 1 * 1 = 0 \end{aligned} \tag{2}$$

$$\begin{aligned} \underline{Class = middle} \qquad \qquad \qquad & 1/3 * 1 * 1 * 1 * 1 * 1 = 1/3 \end{aligned} \tag{3}$$

$$\begin{aligned} \underline{Class = high} \qquad \qquad \qquad & 1/3 * 0.5 * 0 * 1 * 0.5 * 1 = 0 \end{aligned} \tag{4}$$

According to the Naïve Bayes classification the class of the energy consumption would be middle, because it has the largest value.

With the prediction of the energy consumption, the Autonomic Manager can now regulate the process of layer 1. The basic objective of the AM is now to regulate and control the process. The AM will choose the process instance with the lowest predicted energy consumption to guarantee the best energy efficiency of the whole process. For that reason, the AM needs to trigger every single CPS to reach a orchestration of the whole process. However, the introduced methodology only allows to regulate a process instance at the beginning of the process, that means the sequence of an instance will be determined at the start of the processes. If the process instance once started, no adaptations are possible according to the Naïve Bayes classification. Another alternative would be the usage of logistic regression, which usually comes along with more computing complexity.

4 Conclusion

The present paper illustrated an architecture to increase the energy efficiency of a whole industrial process, that consists of Cyber Physical Systems. With the usage of the Naïve Bayes classification it was shown how the energy consumption can be predicted based on process data. The suggested methodology allows companies to optimize their energy efficiency in manufacturing processes and to prepare the way for a sustainable industrial process. However, it is in the charge of organizations how far their industrial processes will be optimized regarding to their energy efficiency. It is imaginable that other factors are also important to guarantee an efficient manufacturing process. For example, to predict failures of activities and processes can also be an important fact for reducing costs and to increase the competitiveness of the company.

For the present paper, the Naïve Bayes classification was chosen to predict the energy consumption based on process data. There are certainly a lot of other

methods, which can be used to predict a continuous or categorical attribute in process data. Some authors for example suggested to use decisions trees, nearest neighbor classification or artificial neuronal networks for prediction based on process data [17, 18, 19]. The present paper provides a first point of contact as well as a basic framework to examine detailed aspects in further research projects.

References

- [1] S. Bode and H.-M. Groscurth, “Zur Wirkung des EEG auf den Strompreis,” HWWA Discussion Paper, Tech. Rep., 2006.
- [2] C. Goebel, H.-A. Jacobsen, V. del Razo, C. Doblander, J. Rivera, J. Ilg, C. Flath, H. Schmeck, C. Weinhardt, D. Pathmaperuma, H.-J. Appellrath, M. Sonnenschein, S. Lehnhoff, O. Kramer, T. Staake, E. Fleisch, D. Neumann, J. Strüker, K. Ereş, R. Zarnekow, H. Ziekow, and J. Lässig, “Energieinformatik: Aktuelle und zukünftige Forschungsschwerpunkte,” *WIRTSCHAFTSINFORMATIK*, vol. 56, no. 1, pp. 31–39, Feb. 2014.
- [3] J. O. Kephart and D. M. Chess, “The vision of autonomic computing,” *Computer*, vol. 36, no. 1, pp. 41–50, Jan. 2003.
- [4] J. A. Lalande, Philippe McCann and A. Diaconescu, *Autonomic computing : principles, design and implementation*. London: Springer, 2014.
- [5] IBM, “An architectural blueprint for autonomic computing.”
- [6] S. J. Oks, A. Fritzsche, and K. M. Möslin, “An Application Map for Industrial Cyber-Physical Systems,” in *Industrial Internet of Things*, ser. Springer Series in Wireless Technology, S. Jeschke, C. Brecher, H. Song, and D. B. Rawat, Eds. Springer International Publishing, 2017, pp. 21–46, dOI: 10.1007/978-3-319-42559-7_2.
- [7] E. A. Lee, “Cyber Physical Systems: Design Challenges,” in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, May 2008, pp. 363–369.
- [8] C. Kaestle, H. Fleischmann, M. Scholz, S. Haerter, and J. Franke, “Cyber-Physical Electronics Production,” in *Industrial Internet of Things*, ser. Springer Series in Wireless Technology, S. Jeschke, C. Brecher, H. Song, and D. B. Rawat, Eds. Springer International Publishing, 2017, pp. 47–78, dOI: 10.1007/978-3-319-42559-7_3.
- [9] NIST, “Framework for Cyber-Physical Systems,” Jan. 2016.
- [10] M. Weth, *Reorganisation zur Prozessorientierung*, ser. 5. Lang, 1997.
- [11] W. M. Van Der Aalst, A. Adriansyah, A. K. A. de Medeiros, F. Arcieri, T. Baier, T. Blickle, J. C. Bose, P. van den Brand, R. Brandtjen, J. Buijs, and others, “Process mining manifesto,” in *Business process management workshops*. Springer, 2012, pp. 169–194.
- [12] D. Höschele, *Process Mining: mit Energiedaten*. Av Akademikerverlag, Jun. 2017.
- [13] M. Pehnt, “Energieeffizienz – Definitionen, Indikatoren, Wirkungen,” in *Energieeffizienz*, M. Pehnt, Ed. Springer Berlin Heidelberg, 2010, pp. 1–34.

- [14] M. G. Patterson, “What is energy efficiency?: Concepts, indicators and methodological issues,” *Energy Policy*, vol. 24, no. 5, pp. 377–390, 1996.
- [15] W. M. Van Der Aalst, *Process Mining: Data Science in Action*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016.
- [16] M. Bramer, *Principles of Data Mining*, 2nd ed. Springer, 2013.
- [17] F. M. Maggi, C. D. Francescomarino, M. Dumas, and C. Ghidini, “Predictive Monitoring of Business Processes,” in *Advanced Information Systems Engineering*, ser. Lecture Notes in Computer Science. Springer, Cham, Jun. 2014, pp. 457–472.
- [18] B. Kang, D. Kim, and S.-H. Kang, “Real-time business process monitoring method for prediction of abnormal termination using KNNI-based LOF prediction,” *Expert Systems with Applications*, vol. 39, no. 5, pp. 6061–6068, Apr. 2012.
- [19] N. Behrooz Khazaei, T. Tavakoli, H. Ghassemian, M. H. Khoshtaghaza, and A. Banakar, “Applied machine vision and artificial neural network for modeling and controlling of the grape drying process,” *Computers and Electronics in Agriculture*, vol. 98, pp. 205–213, Oct. 2013.

5 Appendix A

Table 4. The example data set

Trace	Time Stamp	Activity	Energy Consumption [in Wh]
1	1-12-2017: 11.00	A	50
1	1-12-2017: 12.00	B	30
1	1-12-2017: 12.30	C	60
1	1-12-2017: 12.45	D	10
1	1-12-2017: 13.45	E	35
2	1-15-2017: 9.00	B	30
2	1-15-2017: 10.05	C	55
2	1-15-2017: 10.45	D	25
2	1-15-2017: 12.00	E	25
3	1-17-2017: 15.00	B	25
3	1-17-2017: 15.15	C	65
3	1-17-2017: 17.30	F	80
3	1-17-2017: 18.00	E	35
4	1-23-2017: 9.00	A	45
4	1-23-2017: 10.10	B	40
4	1-23-2017: 10.45	C	55
4	1-23-2017: 12.10	D	15
4	1-23-2017: 13.30	E	35
5	1-23-2017: 10.50	B	30
5	1-23-2017: 13.35	D	20
5	1-23-2017: 13.10	E	35
6	1-30-2017: 11.00	A	50
6	1-30-2017: 12.00	B	50
6	1-30-2017: 13.00	C	60
6	1-30-2017: 13.45	D	10
6	1-30-2017: 14.45	E	35

Eine autonome Optimierung der Stabilität von Produktionsprozessen auf Basis von Prozessdaten

Richard Horn

TU-Dresden

Zusammenfassung. Die Instandhaltung erhält zunehmende Relevanz in industriellen Unternehmen, da sie multiple Bereiche und Ebenen einer Organisation tangiert und wichtige Wettbewerbsvorteile generieren kann. Infolge dessen und auch in Hinblick immer teurer werdender Produktionsanlagen rückt die Instandhaltung in den Fokus unternehmerischer Betrachtung. Das Areal des Autonomic Computing bietet hier ein Anwendungsfeld, welches durch seine Kerneigenschaften hohe Potentiale für eine optimierte Instandhaltungsstrategie birgt. Der vorliegende Beitrag erarbeitet eine Möglichkeit unter Nutzen der zunehmenden Vernetzung der Produktionssysteme eine autonome Steuerung des Produktionsprozesses zu erlangen. Diese wird durch die Verwendung von Process- und Data Mining sowie auf Basis von Prozessdaten hinsichtlich des Fehleraufkommens in Produktionsanlagen optimiert.

Schlüsselwörter: Self-organizing autonomic computing ·Cyber-physical networks ·Maintainability and maintenance ·Industry and manufacturing ·Process control systems ·Process Mining

1 Einleitung

Die traditionelle Aufgabe der Instandhaltung ist es, kaputte Dinge instand zu setzen [1]. Diese archaische Sichtweise von Maintenance findet sich in keiner aktuellen Definition von Instandhaltung wieder. Dies resultiert aus der extremen Relevanz, welche Maintenance in industriellen, produzierenden Unternehmen heutzutage einnimmt, da sie einerseits als hoher Kostenfaktor einen Anteil von 4-12% des Umsatzes verursacht und andererseits anteilig für die Effizienz und die Produktionskapazitäten des Unternehmens verantwortlich ist [2, 3]. Aufgrund dessen sind über die vergangenen Jahre multiple Ansätze, wie die "Condition-based Maintenance" oder auch "Predictive Maintenance" entstanden, welche der Berechnung der bestmöglichen Strategie dienen. In ihrer Gesamtheit kristallisieren sich zwei differente Kategorien der Instandhaltungsstrategien heraus: Reaktive- und Proaktive Vorgehensweisen. Während reaktive Strategien als eher rudimentär zu charakterisieren sind, finden proaktive Verfahren die meiste Anwendung in der Praxis [4]. Als Resultat dieser Vielzahl differenter Vorgehen sowie der kontinuierlichen Forschung ist ein Streben nach der optimalen Lösung

zu erkennen. Dennoch existiert momentan kein optimales Vorgehen im Bereich der Instandhaltung um das Fehleraufkommen zu minimieren bzw. Ausfälle effektiv zu verhindern [5]. Hieraus ist erkennbar, dass neue Methoden bzw. Verfahren gefunden werden müssen, um die Instandhaltung zu optimieren. In den letzten Jahren manifestiert sich der Begriff der Cyber-Physical Systems (CPS), welche als Integrationen von Berechnungen und physischen Prozessen differenter Natur, die aus Berechnungs-, Kommunikations- und Kontrollkomponenten bestehen, zu charakterisieren sind [6]. Demnach besteht eine Beeinflussung der physischen Prozesse durch diese Systeme, wie auch oppositionell der Systeme durch die realen Prozesse. Hieraus ergibt sich ein breites Einsatzspektrum für CPS. Weiterführend entsteht durch die Einführung des Autonomic Computing durch IBM im Jahre 2001 und durch die Beschreibung derartiger Systeme mit humanen Eigenschaften wie beispielsweise *"self-healing"* oder *"self-optimizing"* ein neues Paradigma, welches, in Verbindung mit der Verwendung von CPS, enorme Potentiale für zukünftige Problemstellungen bzw. Technologien wie beispielsweise den Bereich der Instandhaltung bietet [5]. Die Anwendung derartiger Systeme im industriellen Kontext, sowie der verwandte Trend des Industrial Internet of Things bergen mehrere Vorteile in multiplen Bereichen eines Unternehmens [7]. Aufgrund dieser charakteristischen Beeinflussung der realen Welt durch solche Systeme sowie deren Anwendung im industriellen Sektor kann eine Überschneidung dieser Forschungsgebiete angenommen werden.

2 Literaturüberblick

Wie bereits aufgezeigt beschreibt die Instandhaltung ein sehr breit aufgestelltes Areal, welches von kontinuierlicher Forschung getrieben ist. Demnach finden sich hier eine Vielzahl an wissenschaftlichen Publikationen, welche differente Maintenance Strategien deskribieren. Aufgrund der Erkenntnis von Unternehmen, Maintenance als fundamentale Komponente der Organisation zu statuieren, ist festzustellen, dass multiple, unterschiedliche Maintenance Konzepte und Maintenance Policies vorliegen. Insgesamt werden in aktueller Literatur zwölf unterschiedliche Maintenance Konzepte expliziert, für die weiterhin multiple, divergente Durchführungsansätze gefunden werden können [4]. Additional finden sich hier mehrere Normen und Standards, welche alle relevanten Grundbegriffe sowie Eigenschaften von Maintenance darlegen und explizieren. Zu nennen sind bspw. die DIN EN 13306 *"Instandhaltung – Begriffe der Instandhaltung"* sowie die BSI 3811 *"Glossary of maintenance management terms in terotechnology"*.

Ähnlich liegt der Sachverhalt im Bereich der CPS vor, wenn auch, dem Alter des Forschungsgebiets geschuldet, in reduzierter Form. Somit bestehen viele Publikationen, welche neben der Definition und Erläuterung von Grundbegriffen versuchen CPS in differenten Kontexten, wie dem Energiemanagement, dem Prozessmanagement, im medizinischen Bereich etc., anzuwenden. Dennoch befindet sich das Konzept dieser Systeme noch in der Entwicklungsphase, beschreibt allerdings bereits, aufgrund der umfassenden Einsatzfähigkeiten, ein wichtiges Schlagwort in der modernen Industrie [8]. Festzustellen ist, dass sich die Ver-

einigung der beiden Areale in aktueller Literatur teilweise wiederfindet, allerdings meist in Form von Erwähnungen bzw. Ergänzungskapiteln, da der Fokus des Großteils der wissenschaftlichen Publikationen auf der Implementierung von CPS im Manufacturing Bereich liegt und somit Instandhaltung nur ein geringer, wenn auch bedeutender Bestandteil des zu erforschenden Themas beschreibt. Demnach ist zu konstatieren, dass die Verwendung von CPS im produzierenden Gewerbe bekannt ist, allerdings der konkrete Bezug zur Instandhaltung nur partiell vertreten ist.

3 Grundlagen

3.1 Instandhaltung

Der Hauptprozess eines industriellen, produzierenden Unternehmens manifestiert sich in der Transformation des Inputs (Material, Energie und Personal) in den Output der Produktion - das Endprodukt. In Folge der Eingliederung von Instandhaltung in diesen Prozess entsteht sowohl ein additionaler In- als auch Output der Produktion [3]. Dieser Zusammenhang wird in Abb. 1 illustriert.

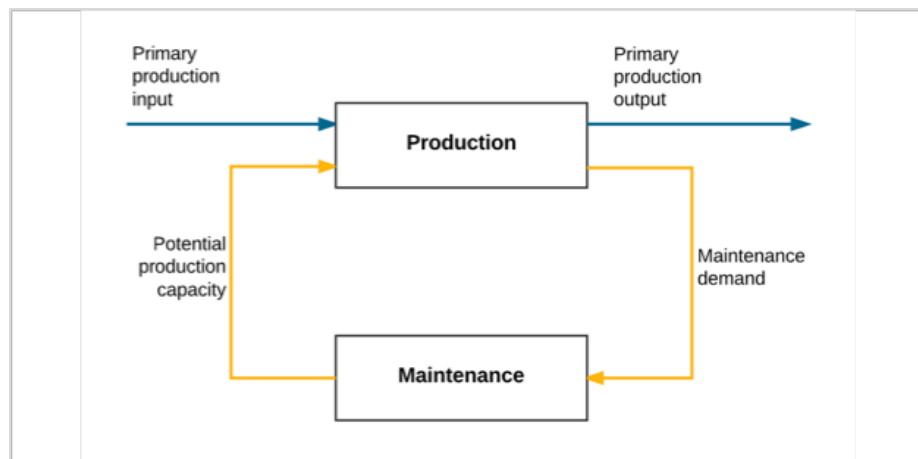


Abb. 1. Zusammenhang von Produktion und Maintenance nach [3]

Die Instandhaltung hat in den letzten Jahren einen enormen Bedeutungszuwachs erfahren, da die Kostendimension in dieser Branche am umfangreichsten ist. Kosten, verursacht durch Instandhaltungsmaßnahmen, können einen hohen prozentualen Anteil an den Gesamtkosten einer Unternehmung einnehmen, so können beispielsweise Kosten in Höhe von 5-15% des Wiederbeschaffungswertes einer Anlage verursacht werden [2]. Aufgrund dessen rückt die Optimierung des Maintenance Managements bzw. der Maintenance Aktivitäten in den Vordergrund unternehmerischer Betrachtung [9].

Zielstellungen der Instandhaltung In ihrer Umsetzung verfolgt die Instandhaltung mehrere Aspekte, welche sie versucht zu erreichen und optimieren. Diese Aspekte werden im Allgemeinen unter dem Begriff der Instandhaltungsfunktion zusammengefasst, welche wiederum in der Literatur differenten Definitionen unterliegt. Die Summe dieser stellt die Gesamtheit der Ziele dar, welche die Instandhaltung verfolgt. So werden ihr alle administrativen und operativen Aufgaben zugesprochen, welche der Aufrechthaltung des arbeitsfähigen Zustands von Produktionsequipment dienen [10]. Ergänzend erklärt die British Standard Institution die Aufgaben zur Wiederherstellung des selbigen Zustandes zu den Zielstellungen [11]. Das European Committee for Standardization wiederum adiiert die Gewährleistung der Anlagenverfügbarkeit sowie durch stabile Systeme, anteilig die Sicherheit des Produktionssystems [12]. Darüber hinaus definiert die DIN EN 13306:2015-09 die Kosteneffizienz, den Umweltschutz sowie die Produktqualität als Ziele der Instandhaltung [13]. Die effiziente Nutzung von Ressourcen sowie die Optimierung des Produktionsequipments im Sinne der Durchlaufzeit und Kapazität werden weiterhin durch [14] und [15] zu den Zielen addiert. Im Gesamten ergeben sich demnach neun Zielstellungen der Instandhaltung, welche in Abb. 2 aufgelistet sind.



Abb. 2. Auflistung der allgemeinen Zielstellung der Instandhaltung

Im vorliegenden Beitrag wird aufgezeigt, inwieweit die Fehlerauftretshäufigkeit sowie die Verlässlichkeit des Produktionsprozesses mithilfe von CPS, basierend auf dem Konzept des Autonomic Computing, minimiert bzw. maximiert werden können. Aufgrund dessen fokussiert dieses Paper primär die Ziele "Erhöhung der Lebensdauer von Produktionsmaschinen", "Minimierung der Ausfallzeiten" sowie "Reduzierung des Ausfall- und Schadenrisikos".

3.2 Process Mining

Grundsätzlich beschreibt ein Produktionsprozess, wie oben bereits aufgeführt, die Transformation des Produktionsinputs in das Endprodukt [3]. Hierbei besteht dieser Prozess aus multiplen Elementen (Maschinen), welche die einzelnen Fertigungsschritte ausführen. Die Abfolge der nötigen Aktivitäten zur Erzeugung des Outputs wird als Prozess definiert und bildet somit alle möglichen Wege der Produktion ab, welche zu jener Transformation führen.

Das Process Mining kristallisiert sich in den letzten Jahren als adäquater Ansatz heraus die realen, operativen Prozessstrukturen zu erkennen und zu analysieren. So beschreibt [16] den Teilbereich der Process Discovery, welcher mithilfe von sog. Process Discovery Algorithmen und auf Basis von Event Logs ermöglicht Prozessstrukturen zu erkennen, welche einen verhaltensrepräsentativen Charakter gegenüber der Realität aufweisen. Weiterhin ermöglicht dies die Erkennung von Prozessvarianten, welche jeweils einen möglichen Ablauf der Produktion repräsentieren, der zu einem Ende dieser führt [16]. Hierfür sind grundsätzlich Event Logs jeder Maschine notwendig, welche einer bestimmten Datenstruktur entsprechen müssen. Der von [17] im Jahre 2014 definierte Standard *"eXtensible Event Stream"* (XES) offeriert hier die Möglichkeit Daten in prozessorientierten Strukturen zu speichern. Hierbei weist das Dokument eine Baumstruktur auf, welche aus einem Wurzelknoten, dem sog. "Log-Objekt", den darunterliegenden "Trace-Objekten", welche die differenten Instanzen des Prozesses darstellen sowie den "Event-Objekten", welche Aktivitäten im Prozess repräsentieren, besteht. Weiterhin wird jedes Event durch multiple Attribute charakterisiert, welche innerhalb des Event-Tags definiert werden. Dabei ist der Standard so konzipiert, dass Erweiterungen im Rahmen des Process Mining implementiert werden können [17]. Eine exemplarische Illustration dieser Struktur findet sich in Abb. 3 wieder.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- XES version 1.0 -->
<log xes:version="1.0" xmlns="http://www.xes-standard.org" xes:creator="Richard Horn">
  <trace>
    <string key="concept:name" value="0"/>
    <string key="variant" value="Variant 1"/>
    <int key="variant-index" value="1"/>
    <string key="creator" value="Richard Horn"/>
    <event>
      <string key="concept:name" value="Aktivitaet1"/>
      <string key="lifecycle:transition" value="start"/>
      <date key="time:timestamp" value="2016-06-02T09:40:44.000+02:00"/>
      <string key="Activity" value="Aktivitaet1"/>
      <string key="org:resource" value="Maschine A"/>
      <string key="Employee" value="Employee12"/>
      <string key="Duration" value="20043"/>
      <string key="Energy_consumption" value="15"/>
      <string key="Material_Amount" value="40"/>
      <string key="Material_cost" value="90"/>
      <string key="StartCost" value="5"/>
      <string key="Total_cost" value="101,02"/>
      <string key="EndCosts" value="106,02"/>
      <string key="ProcessID" value="2"/>
    </event>
    ...
  </trace>
</log>
```

Abb. 3. Illustration der Baumstruktur des XES-Standards

Aufgrund der Fokussierung auf Fehlerereignisse im Produktionsprozess ist an dieser Stelle auf die sog. XES-Lifecycle Extension nach [17] verwiesen. Diese

beschreibt für jede Aktivität ein additionales Attribut, welches Aufschluss über den aktuellen Produktionsstatus gibt. Im Rahmen des vorliegenden Beitrags sind die Status "*start*", "*complete*" und "*pi_abort*" zu erwähnen, wobei letzterer einen Abbruch der gesamten Prozessinstanz beschreibt [17]. Aufgrund des Mangels an Möglichkeiten Fehlerereignisse zu dokumentieren, welche keinen Abbruch der Prozessinstanz zur Folge haben, wird diese XES-Lifecycle Extension um den additionalen Status "*fail_complete*" erweitert, welcher im weiteren Verlauf zur Dokumentation derartiger Fehler dient.

3.3 Cyber-Physical Systems

Cyber-Physical Systems sind unter anderem als Herzstück für alle Änderungen, welche im Rahmen der vierten industriellen Revolution stattfinden, charakterisiert und umfassen in diesem Kontext smartes Produktionsequipment, Speichersysteme und Produktionsanlagen, welche explizit über Fähigkeiten zur wechselseitigen Kommunikation und der Interpretation und Analyse von Daten verfügen [18]. Weiterhin beschreibt [6] diese Systeme als kontextsensitiv und vor allem fähig in der physischen Welt zu agieren. Grundsätzlich bestehen CPS, wie eingangs dargelegt, aus Berechnungs-, Kommunikations- und Kontrollkomponenten, welche sie zu eben genanntem und additional zu Echtzeit-Interaktionen befähigen [19].

Im industriellen Kontext manifestiert sich weiterhin der Begriff der Cyber-Physical Production Systems (CPPS), welcher als das Resultat der Anwendung von CPS in jedem Objekt der Produktionsstraße eines Unternehmens beschrieben ist [20]. Weiterführend sind CPPS anhand von vier Kriterien zu charakterisieren. Zunächst sei hierbei auf die Fähigkeit eines jeden Elements verwiesen, Informationen der umgebenden Objekte aufzunehmen und zu verarbeiten sowie autonom zu agieren. Als weiterer Aspekt nennt [21] die Integration von CPS auf allen Ebenen der Produktion, wodurch sowohl Maschinen wie auch Prozesse und Logistik impliziert sind. Die Interkonnektivität aller Systeme wie auch die Integration des Menschen beschreibt die dritte Charakteristik von CPPS. Das letzte Merkmal beschreibt die Reaktions- und Anpassungsfähigkeit des Systems an externe Einflüsse bzw. Änderungen [21].

3.4 Das Autonomic Computing

Das Autonomic Computing, welches 2001 durch IBM geschaffen wurde, hat in den letzten Jahren stetig steigende Bedeutung in Forschung und Praxis erreicht. Grundsätzlich dient dieses Konzept zur Komplexitätsreduktion von IT-Systemen [22]. Hierbei ist der Name abgeleitet aus der Biologie und referenziert auf das Zentrale Nerven System des Menschen, welches die relevantesten Körperfunktionen, ohne Intervention des Menschen steuert und somit autonom agiert [22]. Resultierend beschreibt das Autonomic Computing Systeme, welche über differente Selbst-Eigenschaften verfügen. Primär beschreibt [23] die Eigenschaft des Selbst-Managements derartiger Systeme als übergeordnete Haupteigenschaft, welche die Eigenverwaltung derartiger Systeme bei zeitgleich minima-

ler Notwendigkeit menschlicher Interaktion beschreibt. Hierbei beschränkt sich das Aufgabenfeld von menschlichen Administratoren darauf, übergeordnete Ziele und Richtlinien für das System zu definieren, während die Durchführung und die technischen Details in der Hand des Systems liegen [23]. Im Rahmen dieser Ausführung von Aktivitäten evaluieren diese Systeme ihre Entscheidung basierend auf einer oder mehreren Situationen, welche sie in ihrer Umwelt erkennen. Folglich ist jegliche Aktion als ein Kontrollzyklus zu charakterisieren, welcher Informationen aus der Umgebung evaluiert um korrekt zu agieren.

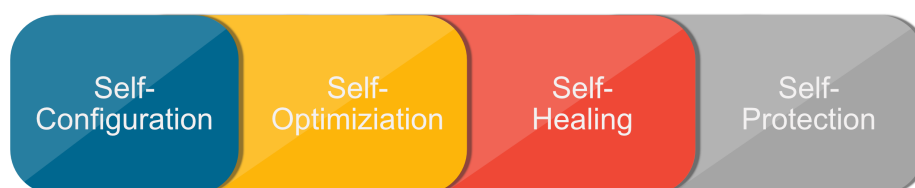


Abb. 4. Die vier Selbst-Eigenschaften eines autonomen Systems

Generell existieren vier differente Self-Eigenschaften (Abb. 4), welche durch diese Kontrollzyklen charakterisiert werden. [22] beschreiben die Typen "*Self-configuring*", "*Self-healing*", "*Self-optimizing*" und "*Self-protecting*". Der erstgenannte Typ beschreibt die Fähigkeit des Systems sich eigenständig und automatisch an wechselnde Umgebungsbedingungen anzupassen. Weiterhin beschreibt die Eigenschaft "*Self-healing*" die Fähigkeit auf Basis eigener Diagnose auf Fehler bzw. Defekte zu reagieren und policy basierte Maßnahmen zu deren Korrektur vorzunehmen. Die dritte Eigenschaft "*Self-optimizing*" wird durch die interne, eigenständige Überwachung und Reallokation von Ressourcen charakterisiert. "*Self-protecting*" hingegen beschreibt die Fähigkeit eines Systems Gefahrensituationen vorherzusehen bzw. zu erkennen und sich durch Eigenanpassung vor diesen Gefahren zu schützen [22, 23].

Der vorliegende Beitrag ist primär auf die Fähigkeit der Selbstkonfiguration sowie der Selbstoptimierung fokussiert, da die Steuerung eines Produktionsprozesses durch ein autonomes System realisiert und hinsichtlich des Fehleraufkommens optimiert wird.

4 Architektur

Die Architektur eines Autonomen Systems ist durch eine Schichtenarchitektur geprägt, wobei die differenten Ebenen mithilfe von Standardmechanismen wie Webservices kommunizieren bzw. interagieren. Hierbei definieren [22] verschiedene Komponenten, welche im Rahmen eines solchen Systems Anwendung finden. Im aktuellen Kontext wird ein zentralisierter Aufbau gewählt und hierfür auf die von IBM 2003 entwickelte Referenzarchitektur MAPE-K (**M**onitor, **A**nalyze, **P**lan, **E**xecute, **K**nowledge) verwiesen. Zunächst sind hierbei die sog. "*managed*

resources” anzusprechen, welche auf unterster Ebene der Architektur die grundlegende Infrastruktur bzw. in Ihrer Gesamtheit den Produktionsprozess abbilden. Generell können diese durch Hard- bzw. Softwarekomponenten verwirklicht sein [22]. Im Kontext eines Produktionsprozesses sind diese Ressourcen durch die Maschinen der Produktion repräsentiert, welche die eigentlichen zu prozessierenden Aktivitäten ausführen und somit in einem Prozesskontext stehen. Weiterhin sind die einzelnen Maschinen als CPS charakterisiert und verfügen über multiple Sensoren, welche Informationen über die Aktivität und deren Ausführung dokumentieren, die an den Autonomic Manager (AM) übertragen werden [24]. Demnach erfüllt die in Abb. 5 illustrierte Architektur den Tatbestand eines ”Cyber-Physical System of Systems” (CPSoS), welches durch eine Kombination einzelner, dezentraler CPS charakterisiert ist und differente Aspekte des Autonomic Computing erfüllt [22]. Weiterhin definiert [25] mehrere Eigenschaften, welche CPSoS von einzelnen CPS unterscheiden. Hierunter zählt beispielsweise die dynamische Rekonfiguration sowie Entwicklung des Gesamtsystems, das teilweise autonome Verhalten der Subsysteme wie auch die verteilte Steuerung und Überwachung des Systems [25].

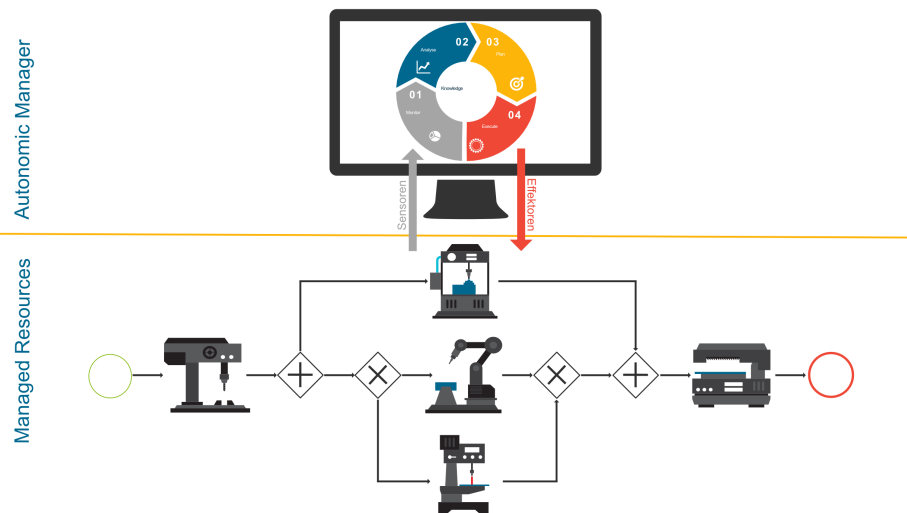


Abb. 5. Architektur des autonomen Systems

Die übertragenen Daten erlauben dem Autonomic Manager den Produktionsprozess zu überwachen und Eingriffe auf Basis dieser Informationen zu realisieren. Der Autonome Manager ist hierbei als Softwarekomponente definiert, welcher, mithilfe der durch die Sensoren der Ressourcen bereitgestellten Daten, Aktionen plant und durchführt und dabei den übergeordneten und von Menschen definierten, globalen Zielen folgt. Demnach wird ein zentralisierter Aufbau gewählt, bei welchem der Autonomic Manager über sämtliche Infor-

mationen der einzelnen CPS verfügt und zusätzlich prozessorientiertes Wissen aufweist was ermöglicht gezielte Eingriffe in die Produktion vorzunehmen. Als letzter Bestandteil sind die Effektoren zu erwähnen, welche Änderungen unterschiedlicher Granularität, initiiert durch den Autonomic Manager, durchführen [24].

5 Funktionsweise

Zur Illustration der Funktionsweise dient folgende Grafik, welche jene ablauforientiert im Standard BPMN 2.0 darstellt. Zunächst ist hier der Startpunkt in den Produktionsmaschinen zu finden, welche jeweils eigene CPS repräsentieren und auf Ausführungsbefehle reagieren. Als weiteren Akteur ist, in Analogie zur soeben präsentierten Architektur, der Autonomic Manager zu erwähnen, welcher jene Befehle erteilt. Zunächst steht im Prozess die Ausführung des eigentlichen Produktionsschrittes im Vordergrund.

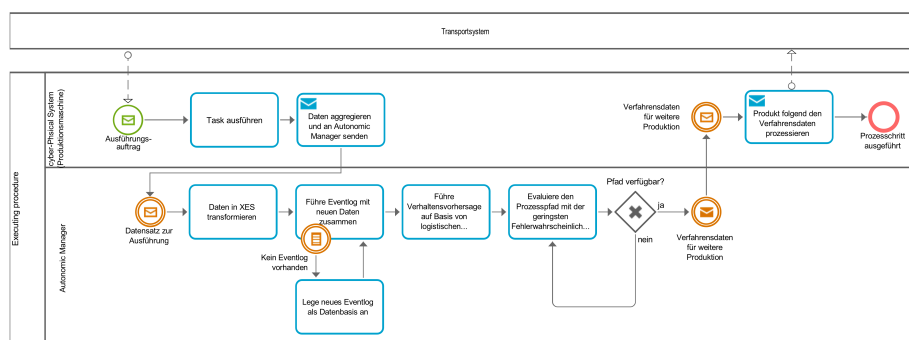


Abb. 6. Illustration der Funktionsweise

Im Rahmen dieser Ausführung dokumentieren die differenten CPS eine Vielzahl an Logfiles, welche im nächsten Schritt an den Autonomic Manager übertragen werden. Weiterführend werden diese Daten innerhalb des Managers aufbereitet und in das XES Format transformiert. Das Zusammenfügen der neuen XES Ausschnitte mit den bereits vorhandenen historischen Daten ermöglicht durch die Anwendung von Process Mining das Erkennen des Gesamtprozesses inklusive seiner Aktivitäten und Varianten. Resultierend erreicht der Autonomic Manager in diesem Zuge prozessorientiertes Wissen, welches im Folgenden Verwendung findet. Generell dienen diese Event Logs demnach als Datenbasis für Entscheidungen des Autonomic Manager.

Resultierend der Fokussierung auf Fehlerereignisse im Produktionsprozess wird im Folgenden ein mögliches Verfahren vorgestellt, um Abhängigkeiten zwischen differenten Fehlern aufzudecken und somit Aussagen über den Potentiellen Verlauf der Produktion zu treffen. Somit kann festgestellt werden, dass aufgrund

eines Fehlers in einer bestimmten Maschine die Fehlerwahrscheinlichkeit einer anderen Maschine in der Produktion steigt. Zur Analyse dieses Sachverhaltes wird auf Basis des Process Mining exemplarisch das Data Mining Verfahren der Logistischen Regression mit dem Ziel der Klassifikation verwendet. Generell ist zu erwähnen, dass hier auch andere Klassifikationsverfahren wie Künstliche Neuronale Netze oder Entscheidungsbäume Anwendung finden können, im vorliegenden Beitrag allerdings der Fokus auf der logistischen Regression liegt. Diese beschreibt die Relation einer Outputvariable und eines Prädiktors und gibt hierbei Aufschluss darüber, mit welcher Wahrscheinlichkeit ein Ereignis der Outputvariable in Abhängigkeit des Prädiktors eintritt [26]. Übertragen auf den aktuellen Kontext beschreibt der Prädiktor den dokumentierten Fehler der Maschine, welche eine Aktivität ausgeführt hat sowie die Outputvariable das Verhalten derjenigen Maschine bzw. desjenigen Prozessschrittes welcher fokussiert wird. Als Resultat dieser Berechnung ergibt sich der sog. $\exp(\beta)$ -Wert ϵ , welcher die Wahrscheinlichkeit für das Auftreten eines Fehlers in der Outputvariable beschreibt. Somit ergibt sich beispielsweise die Erkenntnis: Wenn in Maschine A ein Fehler auftritt, dokumentiert Maschine B ϵ -mal wahrscheinlicher ebenfalls einen Fehler in Referenz zum Zustand eines bisher fehlerfreien Verlaufs der Produktion.

Indem dieses Verfahren bzgl. jeglicher Maschinenkonstellation durchgeführt wird, welche im unmittelbaren, weiteren Verlauf der Produktion vorkommen kann, generiert der Autonomic Manager prädiktives Wissen über den weiteren Produktionsprozess. Dieses erlaubt eine Entscheidung über die weitere optimale Route mit minimaler Fehlerauftrittswahrscheinlichkeit. Wie in Abb. 6 zu erkennen ist, ist der Trigger für diese Analyse und Entscheidung eine Maschine, welche eine Anweisung zur Durchführung einer Aktivität verwirklicht. Resultierend wird die verhaltensorientierte, prädiktive Analyse nach jedem Schritt in der Produktion durchgeführt, wobei die Menge der Prädiktorvariablen bei fortschreitender Produktion ansteigt, sodass die Entscheidungen des Autonomic Managers zu einer optimalen Route durch den Prozess führen. Aufgrund des enormen Rechenaufwandes wird hier im Gesamten ein Greedy-Verfahren verwendet, nach welchem die Berechnungen der Fehlerwahrscheinlichkeiten lediglich mit den unmittelbar folgenden Maschinen bzw. Maschinengruppen als Outputvariablen berechnet wird. Im Anschluss daran, wird der Pfad mit dem geringsten $\exp(\beta)$ -Wert gewählt. Für die Auseinandersetzung mit differenten prozessualen Begebenheiten sei an dieser Stelle auf Kapitel 5.1 verwiesen, welches dieses Themenfeld genauer behandelt.

Grundsätzlich ist hier festzustellen, dass ein Lernprozess des Autonomic Managers zu verzeichnen ist, da dieser auf Basis von historischen Daten und prädiktiven Analysen Entscheidungen trifft. Diese historischen Daten steigen in ihrem Umfang linear zu der Menge an prozessierten Instanzen. Dies resultiert in höherer Präzision der prädiktiven Analysen und somit in sichereren bzw. besseren Entscheidungen. Weiterhin ist in diesem Kontext auf das Konzept der Mustererkennung zu verweisen, welche die, meist automatisierte, Klassifizierung von Objekten auf Basis von subjektiven, berechneten Parametern meint [27]. In Relation zum aktuellen Kontext ist der AM fähig Muster zu erkennen, welche

eine bestimmte Abfolge von Fehlern im Produktionsprozess widerspiegeln und diese entsprechend in Entscheidungen zu berücksichtigen. Logisch geschlussfolgert erlangen diese Patterns ebenfalls mit steigender Instanzanzahl an Relevanz und gewinnen über den Gesamtzeitraum der Verwendung zunehmend an Aussagekraft. Resultierend erfüllt dieses Vorgehen den Tatbestand der Eigenschaft des *"self-optimizing"* autonomer Systeme.

Darüber hinaus ist anzumerken, dass bei Hinzufügen neuer Produktionsmaschinen jeweils mindestens eine Produktionsinstanz über diese Maschine prozessiert werden muss, damit jene in die Datenbasis des Autonomic Manager aufgenommen wird. Somit muss der AM in diesem Falle angelernet werden. Sobald neue Maschinen in den Datenbestand integriert wurden können diese im Rahmen der prädiktiven Analysen und Entscheidungen einbezogen werden.

Gerade in Hinblick auf die Entscheidungsfähigkeit des Autonomic Manager ergibt sich ein weiterer Vorteil. Durch fortführende Analyse des Datenbestandes durch den Autonomic Manager sind Vorhersagen bzgl. der Ausfallzeitpunkte von einzelnen CPS möglich. Auf Basis dessen gibt der AM diese Informationen an Mitarbeiter des Instandhaltungsbereichs weiter und manifestiert sich so als Initiator von Maintenance Aktivitäten. Dieses Vorgehen beschreibt eine direkte Änderung am bestehenden Maintenance Prozess, da die differenten Aufgaben nicht durch Vorgesetzte an Mitarbeiter weitergeleitet werden, sondern das autonome System als Auftraggeber fungiert [28]. Gleichzeitig wird in diesem Fall die vorherstehende Instandhaltungsmaßnahme in der laufenden Produktion berücksichtigt, indem jegliche Prozessvariante, welche die zu wartende Maschine beinhaltet vorübergehend, bis zur Fertigstellung der Wartung, keine Verwendung findet.

Als letzten Schritt im Vorgehen gibt der Autonomic Manager die Daten seiner Entscheidung an das aktive CPS in der Produktion weiter, welches das Produkt mit den dazugehörigen Zieldaten an ein Transportsystem übergibt. Gesamt kann durch dieses Verfahren eine autonome Steuerung des Produktionsprozesses sowie eine Optimierung dieses bzgl. bestimmter Instandhaltungsziele erreicht werden.

5.1 Topologie

Im Folgenden werden drei spezielle, differente Fälle betrachtet, welche verschiedene prozessuale Situationen beschreiben, anhand welcher das Vorgehen der Bewertung vorgestellt werden soll. Generell verfolgt die vorgestellte Lösung eine pessimistische Haltung, da der Umgang mit Fehlerereignissen im produzierendem Gewerbe eine gewisse Sensibilität erfordert, welche aus den eventuell gravierenden Folgen eines Ausfalls resultiert.

Zunächst sei auf das in Abb. 7 illustrierte Beispiel von sequentiellen Aktivitäten verwiesen. In Bezug auf die Bewertung der Pfade berechnet der Autonomic Manager die $\exp(\beta)$ -Werte für alle Aktivitäten, welche sequentiell bis zum nächsten Gateway auftreten. Zur Evaluation des optimalen weiteren Pfades wird eine Min-Max-Regel nach Folgendem Schema angewandt, wobei i jeweils

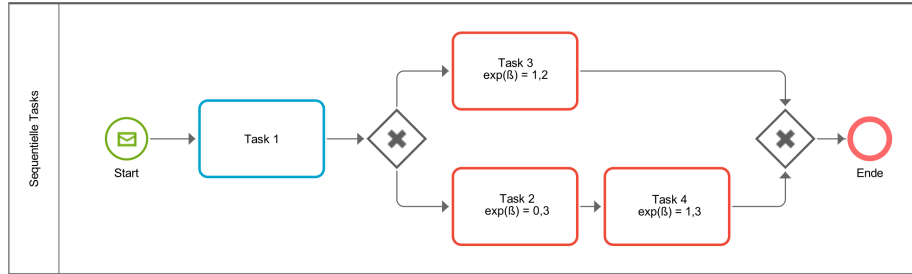


Abb. 7. Fallbeispiel mit sequenziellen Aktivitäten

eine Aktivität der Aktivitätenmenge adressiert:

$$\min(\max(\exp(\beta_i), \exp(\beta_{i+1}), \dots, \exp(\beta_n))) \quad (1)$$

Dies ermöglicht eine Wahl des Pfades, welcher insgesamt die geringste Fehlerwahrscheinlichkeit aufweist, da lediglich das Maximum aller $\exp(\beta)$ -Werte eines Pfades Auskunft über das eventuelle Fehlerrisiko jenes gibt. Die Minimierung der entstandenen Menge an Werten gibt somit den Pfad mit der minimalen Maximal-Fehlerwahrscheinlichkeit an.

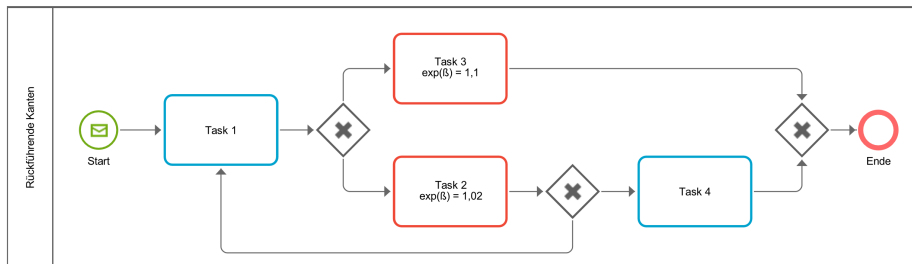


Abb. 8. Illustration eines Beispiels mit Gateways im sequentiellen Kontext

Wie bereits erwähnt arbeitet der Autonomic Manager nach dem Greedy Verfahren, weshalb in dem in Abb. 8 dargestellten Fall eine Bewertung der Fehlerausfallwahrscheinlichkeit von "Task 2" und "Task 3" erfolgt. "Task 4" erhält hier aus Gründen des Rechenaufwandes in realen, komplexen Produktionsprozessen vorerst keine Beachtung. Im vorliegenden Beispiel wird demnach, auf Basis der berechneten Werte, zunächst "Task 2" ausgeführt. Darauf folgend bilden "Task 1" und "Task 2" die Menge der unabhängigen Variablen, welche zur Berechnung der Fehlerwahrscheinlichkeit von "Task 4" sowie erneut "Task 1" verwendet werden. Hierbei werden die unabhängigen Variablen in Interaktion, also gemeinsam, betrachtet, um zu verhindern, dass Prädiktor und Outputvariable identisch sind sowie um die prozessorientierung in die Analyse einzubinden.

Somit basieren die Berechnungen der Ausfallwahrscheinlichkeiten zu jeder Zeit auf allen bisher durchlaufenen Aktivitäten der Instanz.

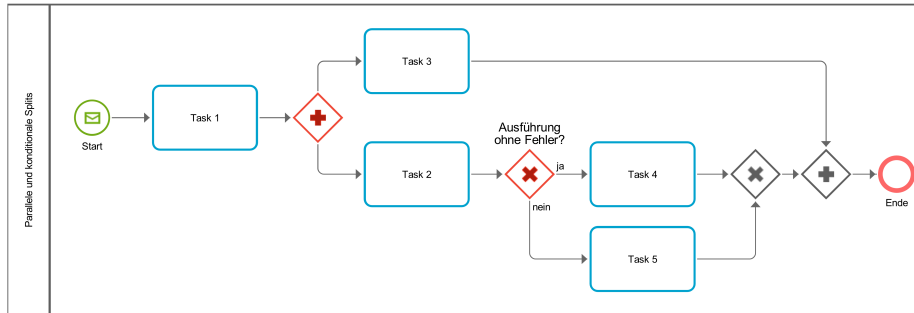


Abb. 9. Illustration eines Beispiels mit parallelem und konditionalem Split

Bezüglich des letzten Falls (Abb. 9) gilt es generell zu erwähnen, dass zwar nach jedem Produktionsschritt eine Datenübertragung an den Autonomic Manager stattfindet, die Vorhersage der Fehlerereignisse sowie das Eingreifen in die Produktion allerdings einer gewissen Optionalität unterliegt. Dies resultiert aus der Nicht-Notwendigkeit des Eingreifens bei parallelen Splits im Prozess, welche implizieren, dass die Ausführung der folgenden Pfade nicht optional ist – diese also in jedem Fall ausgeführt werden. Weiterhin ist auf konditionale Splits in der Produktion zu verweisen, da hier eine Prozedierungsentscheidung auf Basis einer bestimmten Gegebenheit getroffen wird und nicht aus Gründen der Effizienz. Daraus folgt, dass hier ebenfalls kein Eingriff in den Produktionsablauf stattfindet. Generell ist allerdings anzunehmen, dass diese Spezialfälle, aufgrund der hohen Komplexität und dem hohen Maschinenaufkommen, in realen Produktionsprozessen nicht auftreten.

6 Ausblick und Message

Der Instandhaltung wird eine wachsende Relevanz im produzierenden Gewerbe zugesprochen, da sie multiple Bereiche und Ebenen einer Organisation tangiert und wichtige Wettbewerbsvorteile generieren kann. Infolge neuer, fortgeschrittener Produktionstechnologien sowie deren Vernetzung ist eine Veränderung der Instandhaltungsanforderungen in den letzten Jahren zu verzeichnen [29].

Besonders das Areal des Autonomic Computing bietet hier viele Möglichkeiten die zunehmende Vernetzung der Anlagen gezielt und vorteilbringend zu verwenden. Der vorliegende Beitrag stellt eine Methode vor, die Aufgabe der Instandhaltung – die Verlässlichkeit des Produktionsprozesses – durch Autonomic Computing, mithilfe des Process- und Data Mining zu verändern und zu optimieren. Hierbei konnte exemplarisch eine Möglichkeit aufgezeigt und dargestellt werden,

den Produktionsprozess von industriellen Unternehmen durch Autonomic Computing im Rahmen einer Zwei-Schichten Architektur zu steuern. Weiterhin konnte unter Verwendung der Methodik des Process Mining sowie von Data Mining, im Form des Klassifikationsverfahrens der logistischen Regression, eine Methode beschrieben werden, diese Steuerung hinsichtlich des Fehleraufkommens in der Produktionsanlage zu optimieren. Als Basis dieser Steuerung bzw. der Vorhersage dienen Prozessdaten, welche aufbereitet werden und im Rahmen des Process Mining für den Erhalt von Prozessverständnis zuständig sind.

Der vorliegende Beitrag bietet eine Anlaufstelle sowie ein Grundgerüst, um in fortführenden Forschungen feinere, tiefergehende sowie additional Aspekte zu untersuchen.

Literaturverzeichnis

- [1] A. H. Tsang, A. K. Jardine, and H. Kolodny, "Measuring maintenance performance: a holistic approach," *International Journal of Operations & Production Management*, vol. 19, no. 7, pp. 691–715, Jul. 1999.
- [2] A. A. Rasch, *Erfolgspotential Instandhaltung: Theoretische Untersuchung und Entwurf eines ganzheitlichen Instandhaltungsmanagements*. Erich Schmidt Verlag GmbH & Co KG, 2000.
- [3] C. W. Gits, "Design of maintenance concepts," *International Journal of Production Economics*, vol. 24, no. 3, pp. 217–226, 1992.
- [4] M.-H. Karray, B. Chebel-Morello, and N. Zerhouni, "PETRA: Process Evolution using a TRAcE-based system on a maintenance platform," *Knowledge-Based Systems*, vol. 68, pp. 21–39, Sep. 2014.
- [5] H. Song, D. B. Rawat, S. Jeschke, and C. Brecher, *Cyber-Physical Systems: Foundations, Principles and Applications*. Morgan Kaufmann, Aug. 2016.
- [6] C. Kaestle, H. Fleischmann, M. Scholz, S. Haerter, and J. Franke, "Cyber-Physical Electronics Production," in *Industrial Internet of Things*, ser. Springer Series in Wireless Technology, S. Jeschke, C. Brecher, H. Song, and D. B. Rawat, Eds. Springer International Publishing, 2017, pp. 47–78, dOI: 10.1007/978-3-319-42559-7_3.
- [7] S. Jeschke, C. Brecher, H. Song, and D. B. Rawat, Eds., *Industrial Internet of Things*, ser. Springer Series in Wireless Technology. Cham: Springer International Publishing, 2017.
- [8] J. Lee, H. D. Ardakani, S. Yang, and B. Bagheri, "Industrial Big Data Analytics and Cyber-physical Systems for Future Maintenance & Service Innovation," *Procedia CIRP*, vol. 38, pp. 3–7, 2015.
- [9] B. Al-Najjar and I. Alsyouf, "Selecting the most efficient maintenance approach using fuzzy multiple criteria decision making," *International journal of production economics*, vol. 84, no. 1, pp. 85–100, 2003.
- [10] P. Muchiri, L. Pintelon, L. Gelders, and H. Martin, "Development of maintenance function performance measurement framework and indicators," *International Journal of Production Economics*, vol. 131, no. 1, pp. 295–302, May 2011.
- [11] Chartered Institution of Building Services Engineers, Ed., *Maintenance engineering and management: a guide for designers, maintainers, building owners and operators, and facilities managers : CIBSE guide M*. Chartered Institution of Building Services Engineers, 2008.
- [12] European Committee for Standardization, *BUSINESS PLAN CEN/TC 319 - MAINTENANCE EXECUTIVE SUMMARY*. CEN, 2013.
- [13] Deutsches Institut für Normung e.V., Ed., *DIN EN 13306:2015 Instandhaltung - Begriffe der Instandhaltung; Deutsche und Englische Fassung prEN 13306:2015*. DIN EN 13306:2015, Aug. 2015.
- [14] A. Kelly, *Maintenance and its management*. Conference Communication, 1989.

- [15] M. Murray, K. Fletcher, J. Kennedy, P. Kohler, J. Chambers, and T. Ledwidge, *Capability Assurance: A generic model of maintenance*. Maintenance Engineering Society of Australia, 1995.
- [16] W. Van Der Aalst, *Process Mining: Data Science in Action*, 2nd ed. New York, NY: Springer, Apr. 2016.
- [17] C. W. Günther and E. Verbeek, “XES Standard definition,” *Fluxicon Process Laboratories, XES Version*, vol. 1, 2014.
- [18] H. Kagermann, W. Wahlster, and J. Helbig, “Deutschlands Zukunft als Produktionsstandort sichern - Umsetzungsempfehlungen für das Zukunftsjahr 2013,” *Industrie 4.0 - Abschlussbericht des Arbeitskreises Industrie 4.0*, 2013.
- [19] Y. Liu, Y. Peng, B. Wang, S. Yao, and Z. Liu, “Review on cyber-physical systems,” *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 1, pp. 27–40, Jan. 2017.
- [20] N. Galaske and R. Anderl, “Disruption Management for Resilient Processes in Cyber-physical Production Systems,” *Procedia CIRP*, vol. 50, pp. 442–447, Jan. 2016.
- [21] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn, and K. Ueda, “Cyber-physical systems in manufacturing,” *CIRP Annals - Manufacturing Technology*, vol. 65, no. 2, pp. 621–641, 2016.
- [22] IBM Corporation, “An architectural blueprint for autonomic computing,” *IBM White Paper*, 2006.
- [23] P. Lalanda, J. A. McCann, and A. Diaconescu, *Autonomic Computing*, ser. Undergraduate Topics in Computer Science. London: Springer London, 2013, doi: 10.1007/978-1-4471-5007-7.
- [24] M. C. Huebscher and J. A. McCann, “A survey of autonomic computing - degrees, models, and applications,” *ACM Computing Surveys*, vol. 40, no. 3, pp. 1–28, Aug. 2008.
- [25] D. Marron, “What are Cyber-Physical Systems of Systems ?” [Online]. Available: <http://www.cpsos.eu/project/what-are-cyber-physical-systems-of-systems/>
- [26] J. Kopp and D. Lois, *Sozialwissenschaftliche Datenanalyse*. Wiesbaden: Springer Fachmedien Wiesbaden, 2014.
- [27] J. Tohka, “SGN-2506: Introduction to Pattern Recognition,” *Tampere University of Technology, Department of Signal Processing*, 2013.
- [28] B. Bagheri, S. Yang, H.-A. Kao, and J. Lee, “Cyber-physical Systems Architecture for Self-Aware Machines in Industry 4.0 Environment,” *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 1622–1627, 2015.
- [29] L. Swanson, “An empirical study of the relationship between production technology and maintenance management,” *International Journal of Production Economics*, vol. 53, no. 2, pp. 191 – 207, 1997.

Assuring Safety in Autonomous Systems

Christian Rose

Technische Universität Dresden

Abstract. Autonomous motorsport events just like the Roborace presents an opportunity for research to develop and test different research strategies. Furthermore, autonomous motorsport event can pose a danger to all humans around the track like viewers and journalists. In this paper, a novel architecture is presented which allows the supervision of autonomous system in such a motorsport event. Through a layered approach, the architecture allows supervision of the race track. Additionally the architecture in presented in this paper presents a method to validate the proper execution of machine learning systems.

Keywords: Self-organizing autonomic computing · Embedded and cyber-physical systems · Cyber-physical networks · Layered systems · Software safety

1 Introduction

The field of autonomous cars is an active research topic in computer science as well as in other science domains. One important technique employed to enable autonomous cars is the method of machine learning. However, the success of these algorithms depends on the accuracy of the training set to cover all possible situations. A prominent example shows that wrong training data can result in catastrophic failures[3]. Combining this knowledge with the awareness that car crashes can end deadly for the passengers the safe and accurate steering of autonomous vehicles has attracted attention.

This paper will present a novel architecture which will be able to ensure the safety of autonomous cars in the context of the Roborace[1]. The Roborace is a motorsport championship[1] where all participating cars are self-driving. One constraint of these cars is that they should only utilize the information obtained through their sensor systems without direct communication between the cars.

While it might not sound necessary to increase the safety of autonomous cars since they are unmanned and therefore no driver can be harmed, it certainly is important to secure the viewers, track marshalls, and journalist around the racetrack. In the past, several crashes in regular motorsport championships have shown that accidents can be fatal for viewers and drivers alike. Combining the knowledge that machine learning systems can fail and the fact, that crashes in a motorsport event can harm humans around the track[18], introduce the necessity to ensure the safety of such an event. This paper proposes a second independent distributed system which will work as a supervisor for the autonomous

car and its internal systems. The presented architecture enables the supervising system to evaluate the decisions done by the autonomous car and intervene in the case of an emergency. The intervention is done by continuously supervising the track and the internal system of the car. In the course of this paper, the autonomous steering system of the car will be called Machine Learning Based Steering System (MLBSS), the supervising part of the system will be called Algorithmic Based Steering System (ABSS). The supervisor system is composed of different parts, which are explained in depth in Section 5.1. This paper is organized as follows: At first, this paper will give a short overview of other work in this field, after that a few definitions are introduced followed by an examination of crashes triggers of past events. Afterward, based on the examination before, a layered architecture is presented, and its components are described and explained. The paper ends with a conclusion and future work.

2 Related Work

One of the first autonomous systems was Alvin[2] presented in 1995. The architecture of the system already included a combination of several neural networks, each network calculated a score which finally decided which of the neural network would control the car[2]. This system was capable to dynamically re-configure itself, by choosing the correct neural network based on the calculated scores. This was probably one of the first works which have been done on supervising and monitoring of an autonomous vehicle; this was also a step towards including parts of autonomic computing in the architecture of an autonomous system, even though the term "autonomic computing" was not yet introduced. Later IBM put that behavior into a real architecture and gave it the name autonomic computing[16] composed of several attributes such as: "Self-configuration, Self-healing, Self-optimization, Self-protecting"[17]. Later when the autonomous cars got in the reach of real world applicability a lot more research has been to conduct to create more safe and adaptive autonomous cars.

A distributed system similar to the one proposed in this paper was introduced by Kichun Jo et. al. [13]. However, the architecture only focuses on the car internals. This paper proposes a layered architecture which includes the car but also takes cars surrounding and the race track into consideration.

Aside of this, research was done on risk categorization [4, 5] in the context of car traffic. This research has been adapted for the later presented risk estimation function presented in this paper.

Much research has been concentration on the realization of reliable and safe networks for inter car communication [8, 9, 11, 6]. A reliable network communication will not be investigated in this paper. It will be assumed that a reliable network connection is available at all times.

Furthermore, there was much research done on improving the privacy and security of the inter car communication[10, 12]. Since the cars are autonomous without drivers privacy will not be part of this paper.

3 Definitions

In the beginning, we have to set some notation and definitions.

3.1 Cyber Physical System

A cyber physical system (CPS) is a combination of real-time computers and the physical world, where the computer is connected to the physical world through sensors and mechanical devices. Most often these are multiple devices forming a distributed systems. An example of such a system would be the smart grid.[15]

3.2 Cyber-Physical System of Systems

A Cyber-Physical System of System is a combination of several distributed Cyber-Physical System which fulfills several aspects of autonomic computing[17]. It is quite clear that we have a Cyber-physical System on our hands when dealing with autonomous systems. In this paper, a Cyber-Physical System of Systems will be created within the scope of the Roborace. According to the CPSoS-Project, a Cyber-Physical System of System includes additional to the aspects of CPS the following points: ”

- Large, often spatially distributed physical systems with complex dynamics
- Distributed control, supervision, and management
- Partial autonomy of the subsystems
- Dynamic reconfiguration of the overall system on different timescales
- Possibility of emerging behaviors
- Continuous evolution of the overall system during its operation.”[15]

3.3 Machine Learning Based Steering System

Machine Learning Based Steering System(MLBSS) describes the traditional system steering an autonomous system; it is often a combination of ”regular” algorithms and different machine learning based systems.

3.4 Algorithmic Based Steering System

The Algorithmic Based Steering System(ABSS) is an additional distributed system which is developed in the scope of this paper. The ABSS is composed of 3 parts

- **CCU**: *Central Control Unit* - Coordinates cars if a high risk situation occurs
- **TMS**: *Track Monitoring System* - Monitors the race track to identify potential risks
- **IVU**: *In Vehicle Unit* - Monitors the MLBSS, the sensor and estimates the risk

4 Requirement Analysis

Before creating an architecture, the requirements should be analyzed.

First, it has to be defined, at what point the architecture should interfere with the race. As mentioned above one of the important rules for the architecture is to not interfere with the race if it is not necessary. In this paper any imminent crash will be classified as an emergency that has to be stopped immediately, one could argue that it should only interfere if a crash happens near humans like the viewer stands but the moment a crash happens, it is impossible to predict the further outcome, such as the trajectory of parts. So in this paper, any situation resulting in a crash will be used to activate the supervisor and resolve the situation. In this section, a problem-based requirement analysis is made. The problems can be divided into two different categories: Those happening on the track and those inside the car. Therefore in this section first examines the risk on the track after that the problems inside the cars are investigated

4.1 Track Layer Problems

There are several risks for the viewers, race marshals and journalist [18] which are in and around the race track. Those risk can even prove to be deadly [18]. In this section, an overview of situations led to crashes in the history of motorsport is given. One of the frequent reasons for the crashes are objects on the track [19–22], this can either be objects flying in from the outside of the race track, or objects which fall off of damaged cars or even people on the race track. Furthermore, another problematic situation is high velocity often combined with bad weather conditions.[23, 24]. An additional reason for crashes and accident is equipment failure[25–30] of the cars; this includes cars catching on fire due to internal failures. Another possibility is outside influence[31] like persons willingly interfere with the race; this will not be part of the architecture because a person dedicated enough will always find a solution to interrupt the race willingly.

4.2 Car Problems

Besides the risks presented on the race track, a machine learning based systems, such as the steering system in autonomous cars, introduce additional issues. Three scenarios which would result in accidents come to mind. The first being a complete failure of the autonomous system leading to crashes or freezes of the system itself. The second being that a high-risk situation is not correctly recognized by the MLBSS for example when the MLBSS does not recognize an other car or object on the race track. The third being that it identifies the situation correctly but is unable to resolve the situation for example because it is on a collision course with an object and another car is blocking the only escape route.

5 The Architecture

Following the problems presented above a few constraints for the architecture are formulated. Because a software bug could lead to a system reboot or a system freeze, the ABSS should run independently on independent hardware. Furthermore, when the MLBSS is unable to identify the situation correctly, the ABSS should be able to take over the control without the confirmation of the MLBSS. That means that the ABSS has to be in control of a switch, of some kind, which changes the control from the MLBSS to the ABSS.

Of course, the ABSS should not rely solely on the sensor data, because a sensor could fail, that means a different way of getting information such as the own position and the position of other cars should be introduced, this is realized by a GPS module for the position and a permanent connection between all cars.

To enable the ABSS to classify the decision made by the MLBSS a possibility to analyze the MLBSS is needed. The estimation is archived by introducing a risk estimation function, which will be calculated by the ABSS and the MLBSS alike. If the estimated risks are significantly different from each other, the ABSS knows that the MLBSS is unable to identify the situation correctly. The third scenario introduces the necessity of a deeper communication between the cars so that a new route can be negotiated. This, of course, prompts the question of how the system would negotiate with each other and who finally decides on the new path, to avoid any uncertainties a central server is introduced, the Central Control Unit(CCU), which will work as a central decider. While the CCU needs a central server which will make all decisions, several access points can be used to ensure reliable connections and add a failsafe, if there are server CCU-Access point, only one will take care of the decisions. The selection can be made by using the access node which is the closest to the related car, or by choosing randomly. Looking at the problems described in Section 4.1 it is evident that some monitor for the race track has to be introduced. On the following pages an distributed architecture of the ABSS its parts is described.

5.1 Architecture Overview

Following the separation done in Section 4 a two layered architecture is proposed, the architecture is composed of a race track layer and a vehicle layer. The race track layer of the architecture will take care of any incidents on the race track, while the vehicle layer will take care of all events in the car. The layered structure of the architecture can be seen in Figure 1.

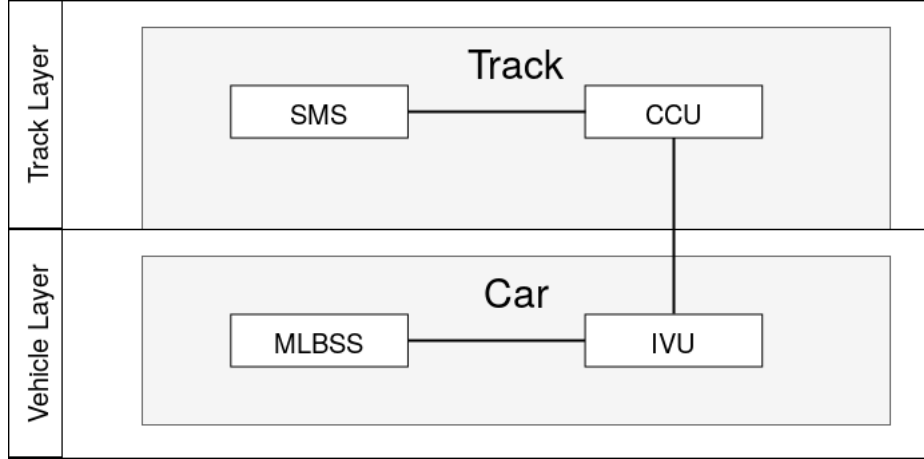


Fig. 1: An overview over the layered architecture

Track Layer: In Figure 2 a visualization of the race track layer can be seen. This layer connects the autonomous cars with the Central Control Unit (CCU) several Track Monitoring Systems (TMS) are installed around the race track and also have a direct connection to the CCU. The TMS contain several sensors to identify potential threads on the race track. Therefore they have to be placed at such a distance from each other, that the whole race track is covered with sensors. The TMS includes heat sensors and cameras. The cameras monitor the race track to see if any objects are detected on the race track, this includes parts which are fallen off of the cars and people running around the race track. The additional heat sensors can detect fire and unusual heat sources which will result in a fire. The CCU unit is connected to all cars at all times during the race. The IVUs will update the position and direction of each car continuously. Furthermore, the TMS will send all race track-related information like people and object on the race track, to the CCU which in turn will send them to the IVU of all cars in the related area. Additionally, the CCU has a complete model of the race track, that way the CCU holds all information relevant to the race track. Combining all this information allows the CCU to control all cars on the race track similar to the way cars are controlled in racing games, allows the supervisor system to avoid hard to verify machine learning based algorithms and therefore makes the algorithms of the CCU more resilient and verifiable.

Vehicle Layer: As mentioned we are working with a two-layered architecture; the second layer is the car itself.

This layer is composed of two separate system, the MLBSS, and the IVU. The IVU's task will be to supervise the MLBSS, send status updates to the CCU and sends any steering information to the cars control. Since the ABSS should be able to take over control at any time, all information from and to the MLBSS

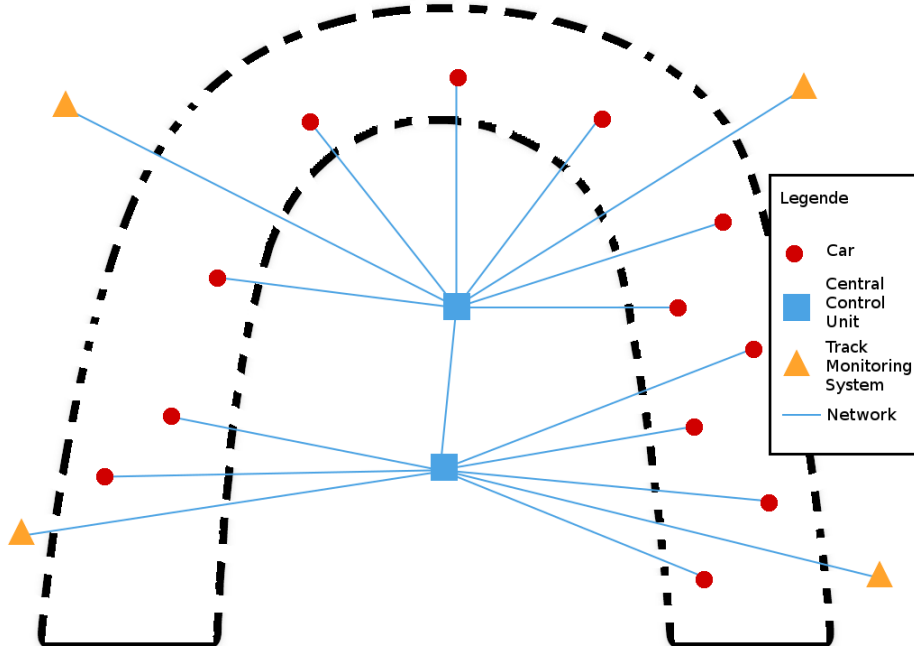


Fig. 2: The architecture on the race track layer

will be passed through the IVU. That way not only can the IVU stop the communication from the MLBSS to the car control at any time, but it can also read all information sent to and from the MLBSS. The IVU should not be able to manipulate the data but can read and process the data to get as much information as possible. The IVU will not steer the car on its own; the steering information will come from the CCU as explained in Section 5.1. The IVU will furthermore validate the MLBSS. Since it is a hard task to supervise a machine learning system by monitoring its internal structure, the IVU will use an indirect method by using a risk estimation. For simplicity, the risk function will use only the direction of the car, the speed the car is traveling at, the direction, and the position and direction of other cars and objects nearby. The risk function will now calculate the risk for the MLBSS, and the ABSS, both pieces of information are sent to the IVU which can compare the results. If the risk estimation of the systems differs significantly, the IVU can be sure, that the MLBSS is failing and will alert the CCU. An example of how such a risk estimator could work is presented in the Section 5.1. The last task of the IVU is to listen to instructions of the CCU, if the CCU sends an incident report, for example, a car nearby is under the control of the CCU, then the IVU should take the control away from the MLBSS and waits for steering instructions of the CCU. The resulting architecture is illustrated in Figure 3 and exemplary algorithms of the IVU is illustrated in Figure 4

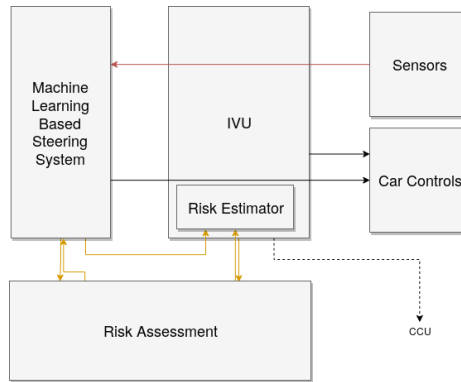


Fig. 3: The Architecture

Figure 4 shows the overall algorithms which decide which system is steering the car.

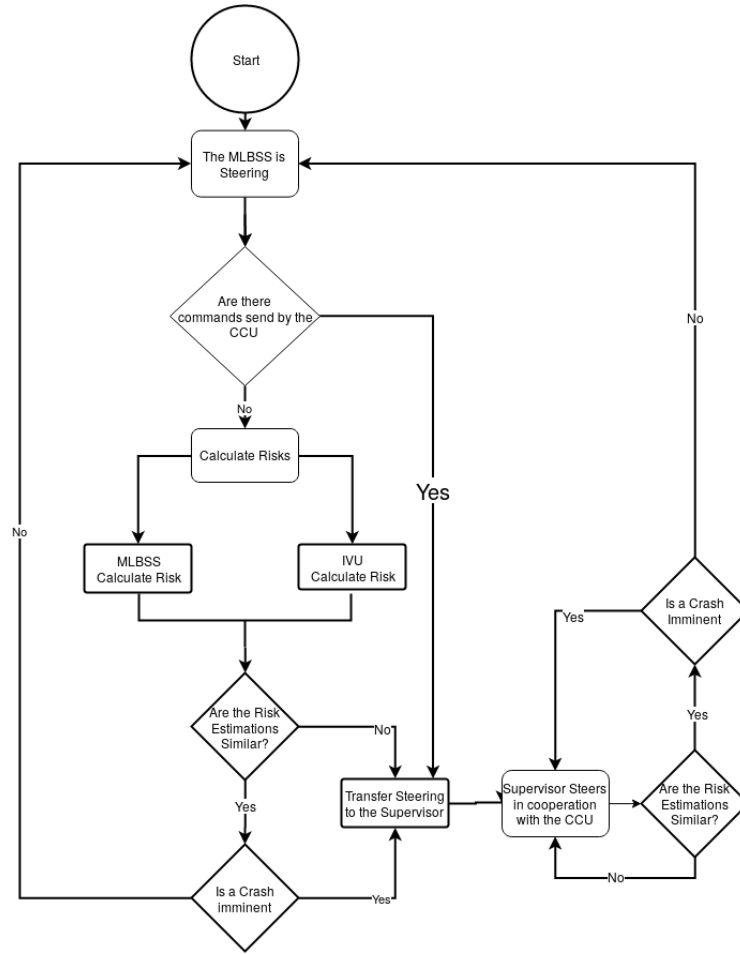


Fig. 4: The algorithms to decide which system is actively steering the car

Bounding Sphere: Assume that one car is in a dangerous situation either on its own or in combination with a second car and the CCU has to defuse the situation. The CCU has to decide on which cars are involved in resolving the conflict. The CCU could take control of only one car, but this could lead to the problem that the other cars nearby could limit the steering options of the first car and therefore the CCU might not be able to resolve the situation. On the other hand, the CCU could take control of all cars on the race track, but this would lead to a significant intervention of the race. The best way would be to choose just a selection of cars which will be controlled by the CCU. A selection of cars could be made by the adaption of a technique normally used in computer games called the "Bounding Sphere"[14]. A Sphere will be set around every car with the middle point set at the position of the car and a radius which, should

be chosen adaptively to the speed the car is traveling at. If said car reports an incident then the control of all cars inside its bounding sphere is handed over to the CCU. To avoid excluding any cars which are close to affected cars but which itself is not inside the bounding sphere, the bounding sphere selection process is propagated through all cars inside the original bounding sphere. The bounding sphere and how they can propagate are visualized in Figure 5.

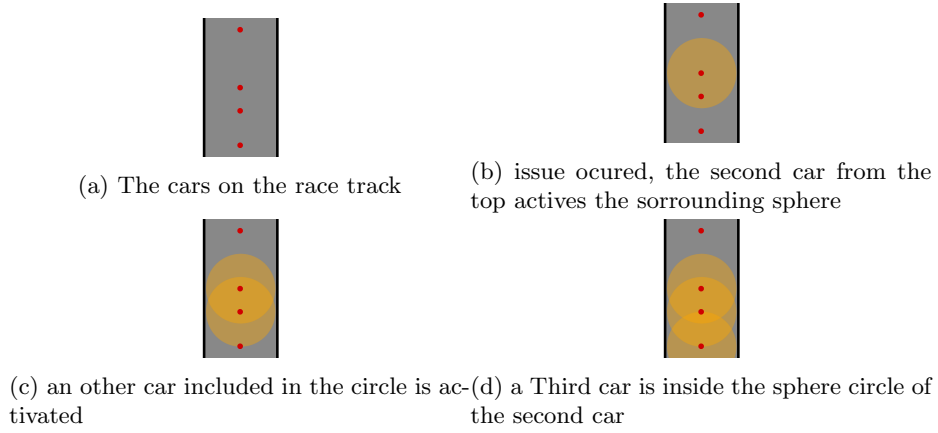
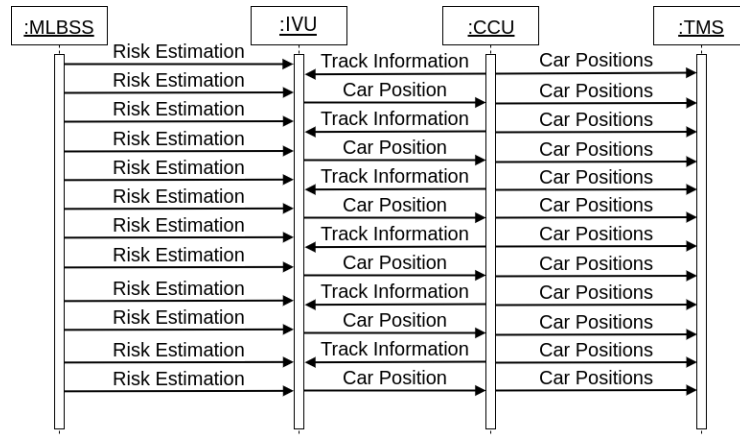


Fig. 5: The forward propagation of the sorrounding spheres

The Message Exchange: A continuous message exchange is happening between the different components of the architecture: The MLBSS and the IVU are in constant contact, and the MLBSS send continuously updated risk estimations to the IVU which in turn will calculate a risk estimation based on its own data. The knowledge of both risk estimations allows the IVU to compare them and that way the IVU can validate the analyses of the MLBSS, this process is explained in further datail in Section 5.1 The IVU sends its current coordinates constantly to the CCU, and the race track information is sent back from the CCU. The car positions are sent by the CCU to the TMS. These continuous connections are illustrated in Figure 6.



In case one of the systems identifies and incident, for example, an imminent crash, additional messages are exchanged. There are two possibilities either the IVU or the CCU identifies an incident. In case the IVU identifies and incident, the IVU switches to controlling mode, it will send a message to CCU here called incident report, the CCU will switch to coordinating mode, will calculate depending on the bounding sphere how many cars have are in a dangerous situation and will send control information to all concerned IVU. The system which has sent the incident report will be the first to send the incident resolved message. If the CCU agrees that there is no incident, it will send a message containing OK, and both will switch its modes back to the original state. Figure 7 illustrates these additional message exchange. If the TMS identifies a possible incident, it will send an Incident Report to the CCU if the CCU agrees that it is indeed an incident, the CCU will forward the report to the IVU. The control informations are sent from the CCU to the IVU as before this time because the CCU initiated the incident, the "Incident Resolved"-message will be sent by the CCU. This sequence diagram of such a situation is illustrated in Figure 8.

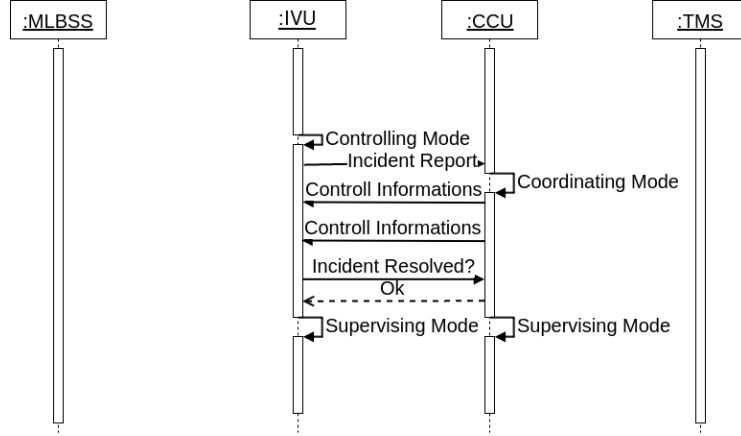


Fig. 7: Symbolizing the message exchange initiated from the IVU

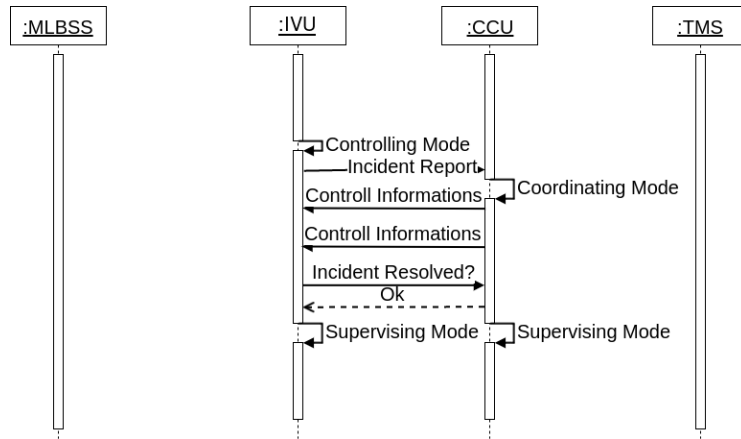


Fig. 8: Symbolizing the message exchange initiated from the TMS

Risk Assessment: An actual risk assessment function is exemplarily explained in this chapter. As the example, the collision between two cars is presented. For the risk assessment function, it is assumed, that a race track can be projected onto a two-dimensional plane, without any intersections. In the case of more complex race tracks, the algorithm could be adapted to a three-dimensional coordinate system. To illustrate the risk assessment a simple risk assessment function should be sufficient, we will adapt the function proposed by Ronald Miller et. al.[7]. Using the position, velocity and the direction which are represented by (x_n, y_n) , (v_n, v_n) and (θ_n, θ_n) respectively while n being a unique identifier for

each car on the course. The following calculation is then done by each car with every other cars in the proximity:

$$x_+ = \frac{(y_k - y_s) - (x_k \tan \theta_k - x_s \tan \theta_s)}{\tan \theta_s - \tan \theta_k} \quad (1)$$

$$y_+ = \frac{(x_k - x_s) - (y_k \tan \theta_k - y_s \tan \theta_s)}{\tan \theta_s - \tan \theta_k} \quad (2)$$

[7] where s is the index for the own car, and $k \in \{1, \dots, n\}$ Next, the Time-To-Cross (TTX) is calculated for each car. After the intersection is calculated, it has to be calculated whether the cars will pass the intersection point at the same time:

$$TTX_1 = \frac{\left| \begin{bmatrix} x_+ \\ y_+ \end{bmatrix} - \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \right|}{|v_1|} * \text{sign}(\left(\begin{bmatrix} x_+ \\ y_+ \end{bmatrix} - \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \right) * |v_1|) \quad (3)$$

$$TTX_1 = \frac{\left| \begin{bmatrix} x_+ \\ y_+ \end{bmatrix} - \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \right|}{|v_2|} * \text{sign}(\left(\begin{bmatrix} x_+ \\ y_+ \end{bmatrix} - \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \right) * |v_2|) \quad (4)$$

[7] The Time-to-React(TTR) is a time set for each car depending on its properties such as braking distance and turning circle, it describes how long a car needs with the current speed, to break or evade an object. The ABSS will take over control and resolve the situation as described in Section 5.1, if the TTX is similar to the TTR, because it can be assumed that the car has to react now, and therefore the MLBSS failed to identify the situation. An illustration of this formula is presented in Figure 9 This formula is easily adaptable for stationary objects such as walls by dividing them into segments and setting the direction to zero, analog it can be used for any object on the race track.

6 Conclusion

Vehicle crashes have proven to be life threatening to its passengers and their surroundings. Furthermore, it has been shown that autonomous cars supported through machine learning are unpredictable, and that makes them genuine dangerous. In this paper, a novel architecture is presented which allows to the indirectly supervision a machine learning algorithms in the context of a autonomous cars. A layered architecture allows the identification of errors inside the car but also any dangers on the race track. The proposed architecture is able to identify and react to the following situation

- Wrong or flawed machine learning algorithms.
- Soft- and Hardware failure of the autonomous algorithm
- Objects, persons or fuel on the race track
- Heat generation inside a car
- Imminent car crashes

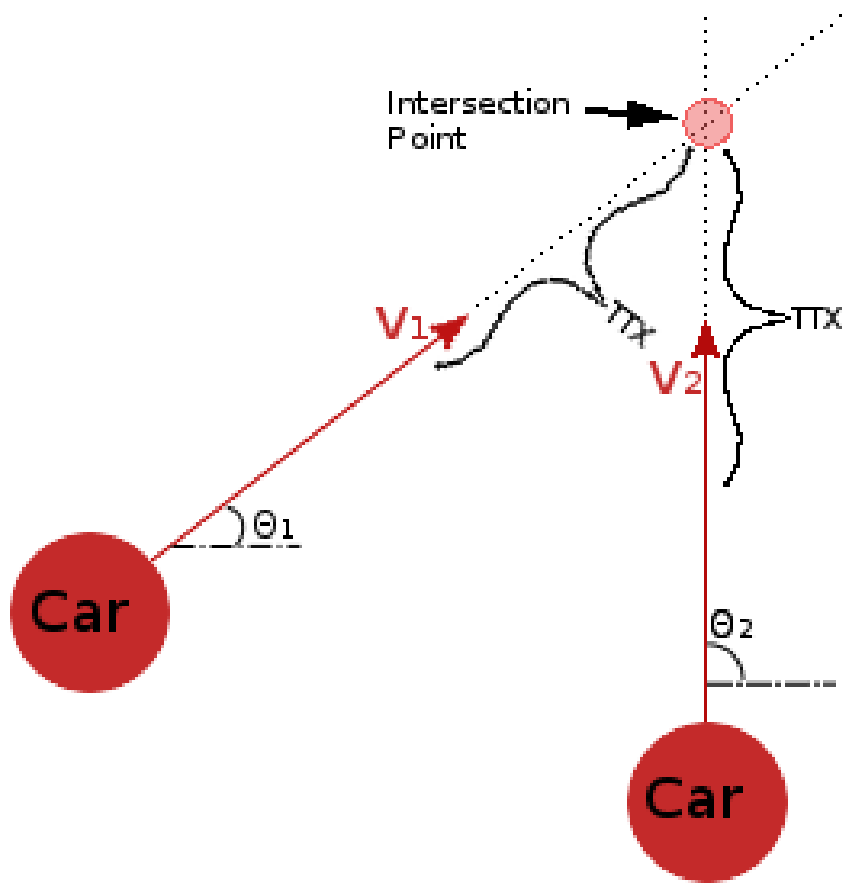


Fig. 9: A illustration of the risk estimation function. Adapted from Ronald Miller et. all[7]

- Control and regulate the traveling speed

Inside the vehicle, a second system is implemented additionally to the traditional Machine Learning Based Steering System (MLBSS). An indirect supervision of the MLBSS based on a risk estimation function allows the second system to identify hardware, software and algorithmic failures. An additional Sensor Network on the race track allows the continuous tracking of anomalies on the race track, including equipment failure. A Central Coordinating System allows the remote steering of cars at any point in time and therefore enables the architecture to cooperative coordinate all cars around obstacles.

Future work could be done on the adaptation of this architecture to regular traffic, conceivable would be to introduce a monitor system for urban areas, including a central CCU for a city. Aforementioned could ensure safe autonomous driving inside an urban area. Further work could be done to move the TMS into the car itself and enable the cars to share their information.

In conclusion, it can be said that this paper presents a comprehensive architecture which could one day avoid life-threatening accidents of autonomous cars.

References

1. Formula E & Kinetik announce driverless support series[Online]. Available:<http://www.fiaformulae.com/en/news/2015/november/formula-e-kinetik-announce-roborace-a-global-driverless-championship.aspx>
2. D.A. Pomerleau, *Neural Network Vision for Robot Driving*, The Handbook of Brain Theory and Neural Networks, M. Arbib, ed. 1995
3. Eliezer Yudkowsky: *Artificial Intelligence as a Positive and Negative Factor in Global Risk*, in Global Catastrophic Risks, Oxford University Press, New York, USA, 2011. ISBN-0199606501
4. Matthias Althoff and Alexander Mergel, *Comparison of Markov Chain Abstraction and Monte Carlo Simulation for the Safety Assessment of Autonomous Cars* in IEEE Transactions on Intelligent Transportation Systems (Volume: 12, Issue: 4), 2011
5. Matthias Althoff et. al., Safety Assessment of Autonomous Cars using Verification Techniques, presented at the American Control Conference, New York, 2007
6. Maxim Raya and Jean-Pierre Hubaux, *Security Aspects of Inter-Vehicle Communications*, published in Sensor, Mesh and Ad Hoc Communications and Networks Workshops, Italy, 2009.
7. Ronald Miller and Qingfeng Huang, *An Adaptive Peer-to-Peer Collision Warning System*, presented at the Vehicular Technology Conference, AL, USA, 2002.
8. R. Stahlmann et. al., *Starting European Field Tests for CAR-2-X Communication: The Drive C2X Framework* presented at Proceedings of 18th ITS World Congress and Exhibition, USA, 2011
9. Subir Biswas et. al., *Vehicle-to-Vehicle Wireless Communication Protocols for Enhancing Highway Traffic Safety* in IEEE Communications Magazine (Volume: 44, Issue: 1) p 74 - 82, 2006
10. Hagen Stübinger et. al. : *sim^TD: A Car-to-X System Architecture for Field Operational Tests* in IEEE Communications Magazine (Volume: 48, Issue: 5), 2010
11. M. Bechler et. al. *Evolving The Europa Its Architecture for Car-To-X Communication*, January, 2009
12. Hagen Stübinger et. al. *A Two-Stage Verification Process for Car-to-X Mobility Data based on Path Prediction an Probabilistic Maneuver Recognition* presented in Vehicular Networking Conference (VNC), Netherlands, 2011
13. Kichun Jo et. al. *Development of Autonomous CarPart I: Distributed System Architecture and Development Process* published in IEEE Transactions On Industrial Electronics, Vol. 61, No. 12, December, 2014
14. David Scherfgen: *3D-Spiele-Programmierung mit DirectX 9 and C++, 3rd edition*, Carl Hanser Verlag Muenchen, Wien, 2006
15. What are cyber physical systems of systems[Online]. Available: <http://www.cpsos.eu/project/what-are-cyber-physical-systems-of-systems/>
16. An architectural blueprint for autonomic computing.[Online]. Available: <http://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf>
17. Philippe Lalanda et. al. *Autonomic Computing: Principles, Design and Implementation*, Springer, 2013
18. David Evans. (2017, January 20). Spectator dies after Hayden Paddon's Monte Carlo Rally WRC crash[Online]. Available: <http://www.autosport.com/news/report.php/id/127810>

19. Motorsport Memorial[Online] Available:<http://motorsportmemorial.org/focus.php?db=ct&n=2984>
20. Motorsport Memorial[Online] Available:<http://motorsportmemorial.org/focus.php?db=ct&n=7198>
21. Motorsport Memorial[Online] Available:<http://www.motorsportmemorial.org/focus.php?db=ct&n=4223>
22. Motorsport Memorial[Online] Available:<http://www.motorsportmemorial.org/focus.php?db=ct&n=6781>
23. Giles Richards. (2014, October 5), Jules Bianchi crash sends F1 into another period of self-examination[Online] Available:<https://www.theguardian.com/sport/blog/2014/oct/05/jules-bianchi-crash-f1>
24. Motorsport Memorial[Online] Available:<http://www.motorsportmemorial.org/focus.php?db=ct&n=1683>
25. Motorsport Memorial[Online] Available:<http://motorsportmemorial.org/focus.php?db=ct&n=3001>
26. Motorsport Memorial[Online] Available:<http://www.motorsportmemorial.org/focus.php?db=ct&n=788>
27. Motorsport Memorial[Online] Available:<http://www.motorsportmemorial.org/focus.php?db=ct&n=123>
28. Motorsport Memorial[Online] Available:<http://www.motorsportmemorial.org/focus.php?db=ct&n=6572>
29. Motorsport Memorial[Online] Available:<http://www.motorsportmemorial.org/focus.php?db=ct&n=6572>
30. Motorsport Memorial[Online] Available:<http://www.motorsportmemorial.org/focus.php?db=ct&n=1503>
31. Motorsport Memorial[Online] Available:<http://www.motorsportmemorial.org/focus.php?db=ct&n=5128>

MAPE-K in der Praxis - Grundlage für eine mögliche automatische Ressourcenzuweisung in der Cloud

Michael Schneider

TU-Dresden

Zusammenfassung. Heutzutage gibt es viele theoretische Ansätze und Referenzarchitekturen. Dabei weicht die Praxis oftmals deutlich ab und die theoretischen Architekturen werden wenig oder gar nicht genutzt. In diesem Paper soll deshalb gezeigt werden, wie die MAPE-K Referenzarchitektur des Autonomic Computing Anwendung im Cloud Computing findet und wo es zu Abweichen von der Architektur kommt. Dazu wird unter Anderem eine kurze Einführung ins Cloud Computing sowie das Autonomic Computing gegeben. Die Überprüfung inwieweit die Theorie in der Praxis Anwendung findet wird anhand Aneka als Beispiel für ein Cloud Framework mit automatischer Ressourcenzuweisung vorgestellt. Kern der Arbeit ist es, den Ablauf einer automatischen Ressourcenzuweisung in Aneka vorzustellen. Kein Ziel dieser Papers ist es, die zur Ressourcenoptimierung verwendeten Algorithmen vorzustellen und die einzelnen Aneka-Komponenten in die MAPE-K Architektur einzuordnen. Diese sind in der Regel auch nicht öffentlich einsehbar und werden von den Cloud Providern geheim gehalten.

Schlüsselwörter: Cloud computing, Self-organizing autonomic computing

1 Einleitung

Cloud Computing ist in der heutigen Zeit eine sehr häufig von Unternehmen verwendete Technologie, um Anwendungen bereit zu stellen. Diese Attraktivität kommt daher, dass die Firmen in durch die Benutzung der Cloud sowohl Kosten, als auch Administrationsaufwand einsparen. Es muss nur das bezahlt werden, was tatsächlich genutzt wird und der Wartungs- und Administrationsaufwand für Hardware ist quasi nicht vorhanden. Ein weiterer, wenn nicht sogar der ausschlaggebende, Vorteil für die Nutzung der Cloud stellt das sogenannte auto-scaling dar. Das bedeutet, dass eine Anwendung bei Bedarf mehr Ressourcen wie zum Beispiel CPUs oder Arbeitsspeicher zugeteilt bekommt, um dadurch eine höhere Last schneller zu bewältigen. Dementsprechend werden von der Anwendung in Zeiten geringer Auslastung auch weniger Ressource genutzt und damit Kosten eingespart. Diese dynamische Zuteilung von Ressourcen ist jedoch nur durch ein komplexes System beim Cloud-Anbieter möglich.

Dieses komplexe System nutzt in der Regel Autonomic Computing als Grundlage zur Selbstoptimierung. Die Optimierung wird dabei durch eine automatisierte Ressourcenzuteilung vorgenommen. Ziel der Optimierung ist an oberster Stelle die Zufriedenheit der Nutzer. Das bedeutet, dass den Nutzern eine Zusage über die Performance des Systems gemacht wird. Des Weiteren sollen die Kosten für den Kunden minimiert werden, also es sollen nur die Ressourcen benutzt und bezahlt werden, welche auch wirklich benötigt und benutzt werden. Zweites Ziel der Optimierung sind die Kosten für den Cloudbetreiber. Aus seiner Sicht sollen Energie und damit auch Kosten gespart werden indem nur die minimal benötigten Ressourcen (also Server) aktive sind.

2 Grundlagen

2.1 Cloud Computing

Cloud Computing wird nach [1] als ein Modell definiert, welches auf Abruf Netzwerkzugriff auf eine Menge von geteilten, konfigurierbaren Ressourcen wie Netzwerk, Speicher und Dienste bietet. Die Bereitstellung dieser erfolgt sehr schnell und mit minimalem Managementaufwand beziehungsweise minimaler Interaktion mit dem Provider.

Die Architektur von Cloud Systemen lässt sich als eine Schichtenarchitektur mit 4 Schichten beschreiben (siehe Abbildung 1). Als unterste Schicht wird

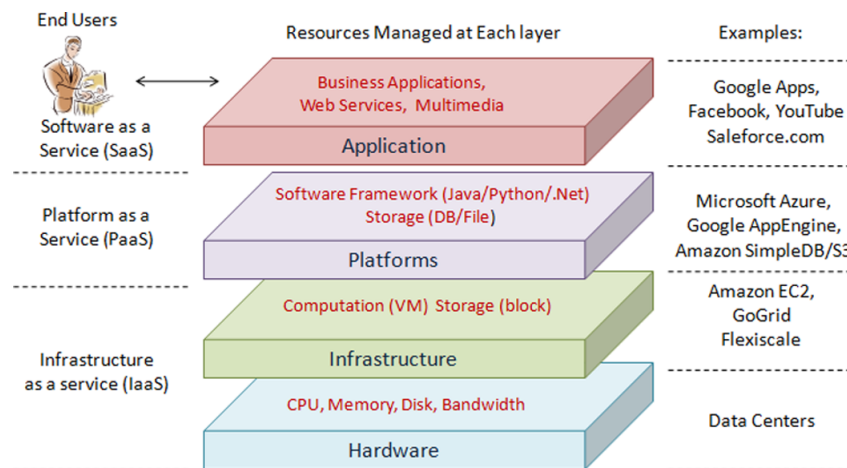


Abb. 1. Cloud Architektur [2]

die Hardware gesehen. Sie ist für das Management der physischen Ressourcen wie Server zuständig. Die darüber liegende Schicht ist die Infrastruktur oder

auch Virtualisierungsschicht. Sie ist für die Generierung von Pools von Speicher und Rechenressourcen zuständig. Ohne diese Schicht und die stattfindende Virtualisierung wäre keine dynamische Zuweisung von Ressourcen möglich. Die Schicht ist deswegen essenziell für das Cloud Computing. Die nächste Schicht ist sogenannte Plattform-Schicht. Auf ihr sind das Betriebssystem und Anwendungsframeworks zu finden. Sie hat als Ziel ein leichteres Deployment von Anwendungen. Ein Beispiel für ein auf dieser Schicht zu findendes Anwendungsframework ist die Google App Engine¹. Die oberste der vier Schichten bildet die Anwendungsschicht. Diese stellt die eigentlichen Anwendungen für die Nutzer zur Verfügung.

Die einzelnen Schichten sind dabei lose gekoppelt, was eine separate Entwicklung der Schichten ermöglicht. Dadurch kann eine breite Masse an Anwendungen mit reduziertem Verwaltungs- und Wartungsaufwand in die Cloud ausgelagert werden.

Für Anwender resultieren aus dieser Schichtenarchitektur 3 Kategorien in welche Cloud Anwendungen eingeteilt werden können. Diese 3 Kategorien sind Infrastructure-as-a-Service (IaaS), Platform-as-a-Service(PaaS) und Software-as-a-Service(SaaS), welche die 3 obersten Schichten der Architektur repräsentieren und dem Nutzer die jeweiligen Funktionalitäten zur Verfügung stellen (siehe Abbildung 1). Cloud Systeme können dementsprechend auch auf anderen Cloud Systemen aufbauen. Beispielsweise kann ein SaaS-System auf eine PaaS System aufgebaut sein, welches wiederum auf einem anderen IaaS-System, basiert.

Durch diese Schichtenarchitektur beziehungsweise die Einteilung in Kategorien wird eine Art Separation of Concerns geschaffen, sodass sich jede Schicht nur um die Schichtspezifischen Belange kümmern muss. Es ist außerdem eine Dienstorientierung möglich und darauf basierend ein Service Management welches auf Service-Level-Agreements(SLAs) basiert. SLAs erhalten dabei wichtige Bestimmungen und Zusicherungen, welche der Dienstanbieter (also der Cloud-Provider) dem Kunden macht, im Bezug auf mögliche Optimierungen und Anpassungen. Ein Beispiel dafür ist die Zusicherung einer bestimmten Antwortzeit welche einen festgelegten Schwellwert nie überschreitet.

Um diese Zusicherungen einzuhalten ist es notwendig, dass der Anwendung automatisch bei Bedarf weitere Ressourcen zugeteilt sowie entzogen werden können. Interessant ist dabei, wie die SLAs auf Low-Level-Anforderungen wie Rechenleistung und Speicher abgebildet werden können. Außerdem sollten die Reaktionszeiten für die notwendigen Optimierungen möglichst niedrig sein. Die Umsetzung dieser Optimierung wird im Folgenden noch weiter diskutiert und erläutert.

Die grundlegenden Schritte für alle existierende Ansätze sind in Abbildung 2 dargestellt.

Das zugrundeliegende Anwendungsperformanzmodell kann dabei unterschiedlich generiert werden. Als Möglichkeiten seien hier nur Kontrolltheorie und Statistical Machine Learning genannt.

¹ <https://cloud.google.com/appengine/>

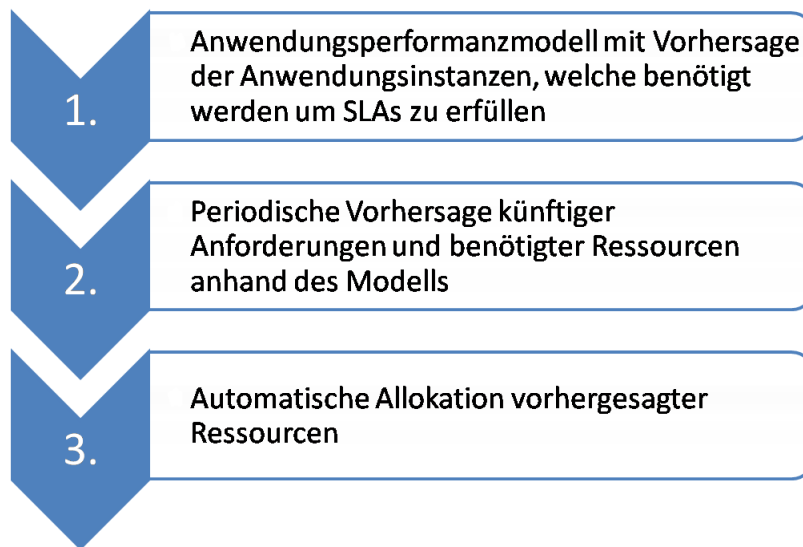


Abb. 2. Grundlegende Optimierungsschritte

2.2 Autonomic Computing

Der Begriff Autonomic Computing wurde 2001 von IBM benutzt und eingeführt [3]. Ein autonomes System wird dabei beschrieben als ein System, welches seine Umgebung wahrnehmen kann, ihr Verhalten modelliert und Aktionen durchführen kann um sein Verhalten oder die Umgebung zu verändern. Im Weiteren hat ein solches System die 4 grundlegenden Eigenschaften Selbstkonfiguration, Selbstheilung, Selbstoptimierung und Selbstschutz [4]. In [4] wird außerdem die Referenzarchitektur eines autonomen Systems beschrieben. Dieses ist aus 5 Blöcken aufgebaut, welche folgend kurz vorgestellt werden.

Der autonome Manager stellt die Implementierung für die Automatisierung der Managementfunktionen bereit. Außerdem implementiert er die sogenannte MAPE-K Kontrollschleife (siehe Abbildung 3). Die Monitoring-Komponente dieser Schleife ist für das Sammeln, aggregieren und Filtern von Details verwalteter Ressourcen zuständig. Beispiele für solche Details sind Metriken wie etwa Antwortzeiten oder Aufrufzahlen von Webanwendungen. Die Analyse-Komponente implementiert Mechanismen zur Analyse von Situationen und der Entscheidung, ob Aktionen zur Zielerfüllung notwendig sind oder nicht. Sie modelliert des Weiteren komplexe Situationen aus diesen Daten und kann Vorhersagen treffen sowie Lernen und damit die Vorhersagen verbessern. In der Planungs-Komponente werden basierend auf der Analyse Aktionen generiert oder bestehende Workflows zur Ausführung gewählt um gesetzte Ziele zu erreichen. Diese Ziele können beispielsweise in vom Provider mit dem Nutzer vereinbarten SLAs definiert sein. Die in der Planungs-Komponente erzeugten Aktionen werden anschließend von der Ausführungs-Komponente (Execution) eingeplant und ausgeführt. Diese 4

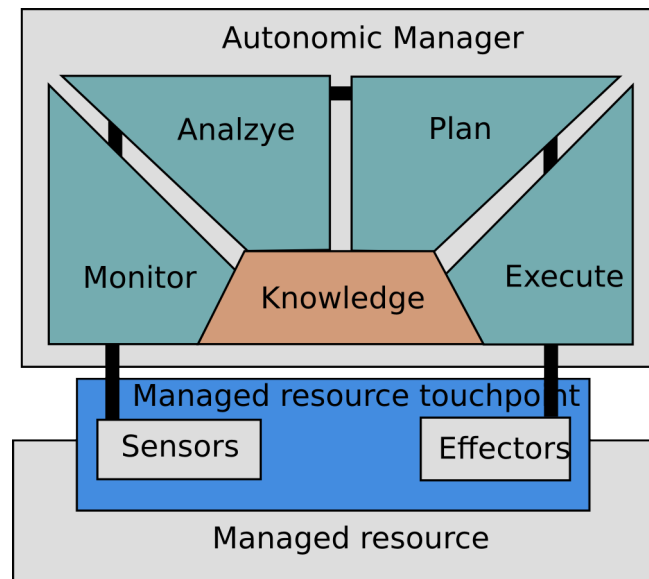


Abb. 3. MAPE-K Kontrollschleife Quelle: https://wwwvs.cs.hs-rm.de/vs-wiki/images/0/01/OpenStack_MAPE-K-Loop.png

Komponenten nutzen für die Entscheidungen über notwendige Aktionen eine gemeinsame Wissensquelle, welche einen eigenen Baustein eines autonomen Systems darstellt. Sie kann während des Schleifendurchlaufs jederzeit von den Komponenten aktualisiert werden. Mögliche Wissensquellen sind beispielsweise Logs welche gegebenenfalls in eine für die Wissensquelle einheitliche Repräsentation transformiert werden müssen.

In der soeben vorgestellten MAPE-K Kontrollschleife lässt sich auch der im vorangegangenen Abschnitt am Ende vorgestellte generelle Ansatz zur Ressourcenoptimierung erkennen. Das Anwendungsperformanzmodell ist in der Knowledge-Komponente enthalten und wird in der Analyse genutzt und gegebenenfalls während des Schleifendurchlaufs aktualisiert. Die Monitoring-Komponente sorgt für die periodische beziehungsweise dauerhafte Überwachung der Ressourcen und ist bei Bedarf der Auslöser für die Analyse. Der dritte Schritt, das Allokieren der benötigten Ressourcen wird durch die Ausführungskomponente erledigt.

Ein weiterer Baustein sind Touchpoints. Sie bilden die Schnittstellen und bieten den Status und Management-Operationen für eine Ressource im System an. Die Kommunikation geschieht hier über Standardschnittstellen in Form von Sensoren und Effektoren der Ressourcen. Sensoren und Effektoren sind entsprechend verknüpft, sodass eine durch den Effektor ausgelöste Änderung im Sensor sichtbar ist. Mithilfe von Sensoren und Effektoren besteht auch die Möglichkeit der Komposition von autonomen Managern.

Manuelle Manager bilden ebenso einen wichtigen Baustein in autonomen Systemen. Sie implementieren die Nutzerschnittstelle, welche es einem Adminis-

trator ermöglicht manuelle Management Funktionen durchzuführen. Manuelle Manager arbeiten oftmals mit autonomen Managern zusammen, indem sie Aktionen delegieren.

Der fünfte Baustein ist der Enterprise Service Bus, welcher die Integration der einzelnen Bestandteile gewährleistet und für die Kommunikation zwischen diesen zuständig ist.

3 Anwendung am Beispiel dynamischer Ressourcenzuteilung mit Aneka

3.1 Aneka

Aneka² ist eine Plattform und ein Framework um verteilte Anwendungen in der Cloud zu entwickeln. Eine Aneka basierte Cloud kann als Sammlung von physischen und virtualisierten Ressourcen angesehen werden, welche über ein Netzwerk verbunden sind [5]. Jede dieser Ressourcen enthält eine Instanz eines Aneka Containers. Dieser repräsentiert die Laufzeitumgebung der verteilten Anwendungen und stellt die grundlegenden Managementfunktionen eines Knotens zur Verfügung, während er alle anderen Operationen an den gehosteten Dienst auslagert. Die Dienste werden hierbei unterteilt in Fabric Services, Foundation Services und Execution Services. Die Interaktion mit dem der Ressource an sich findet ausschließlich in den Fabric Services über die Plattform-Abstraktions-Schicht statt. Die Foundation Services stellen das Kernsystem der Aneka Middleware dar und stellen die Grundfunktionen zur Ausführung spezieller Aufgabemengen bereit. Die Ausführung und deren zeitliche Planung der eigentlichen Anwendungen der Cloud wird in den Application Services geregelt. Die soeben beschriebene Architektur ist in Abbildung 4 visualisiert.

3.2 Dynamische Ressourcenzuteilung in Aneka

Wie aus der Beschreibung der Architektur hervorgeht, ist die dynamische Zuteilung und Integration von Ressourcen zu Anwendungen ein wichtiger Bestandteil von Aneka. Da es sich bei Aneka um ein Framework der Kategorie Infrastructure-as-a-Service handelt, handelt es sich bei den Ressourcen in der Regel um Hardwareknoten in der Form von virtuellen Maschinen. Es spielt dabei keine Rolle, ob diese durch andere IaaS-Provider wie Amazon EC2³ oder lokale Manager wie Xen Server⁴ bereit gestellt werden [6].

Im Normalfall ist Aneka so konfiguriert, dass eine ausreichende Anzahl an Knoten für eine Anwendung zur Verfügung stehen. Im Fall einer außergewöhnlich hohen Last kann durch dynamische Ressourcenzuteilung ein Ausfall beziehungsweise eine Verlangsamung der Anwendung und damit möglicherweise eine Verletzung der vereinbarten Service-Level-Agreements verhindert werden. Dabei wird

² <http://www.manjrasoft.com/products.html>

³ <https://aws.amazon.com/de/ec2/>

⁴ <https://xenserver.org/>

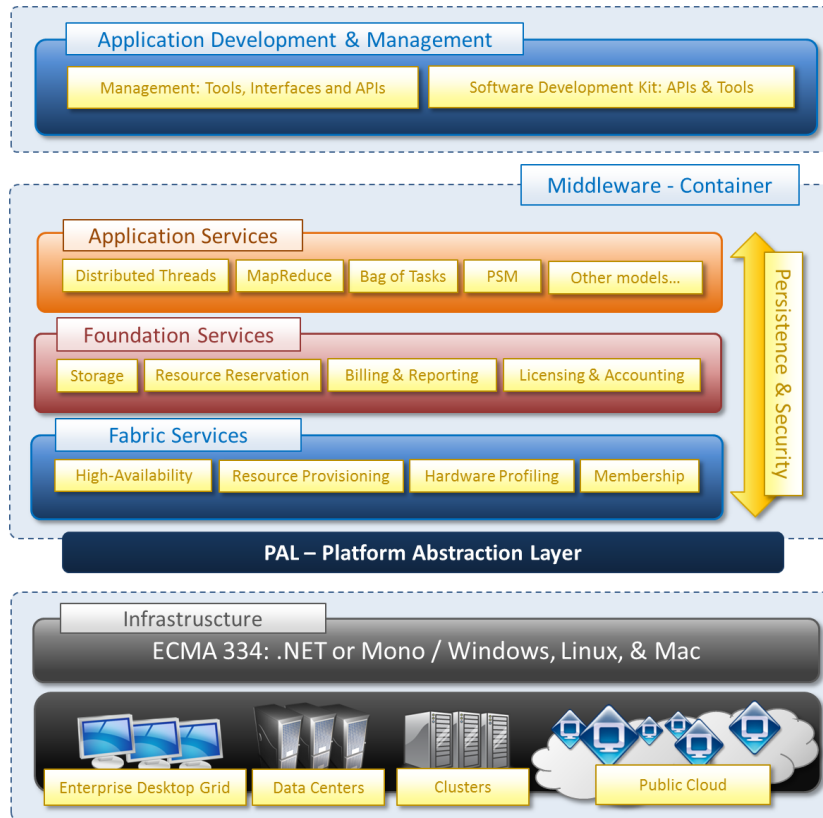


Abb. 4. Aneka Architektur Quelle: <http://www.manjrasoft.com/images/anekaplatform.png>

aus einem Pool ungenutzten Reserveressourcen die entsprechende Anzahl an Knoten zum Erfüllen der SLAs integriert. Um dadurch die Leistung des Systems zu erhöhen müssen mehrere Komponenten des Systems ihre Aktivitäten koordinieren [6]:

- Lastausgleich und Monitoring Facility überwachen die Systemlast und aktuell eingesetzten Knoten.
- Planungs- und Management Facility erkennen den Bedarf an zusätzlichen Ressourcen und entscheiden diese bereitzustellen.
- Budget Management Facility stellt Informationen bereit, ob das System die benötigten Ressourcen nach den Anforderungen des Nutzers, seinem Budget und den Systemeinstellungen zur Verfügung stellen kann.
- Dynamische Bereitstellungs Facility übernimmt die technische Umsetzung der Bereitstellung und überwacht die Freigabe der virtuellen Instanzen

Die Infrastruktur der dynamischen Ressourcenzuteilung in Aneka ist in Abbildung 5 zu sehen. Der in diesem Überblick dargestellte Scheduling Service

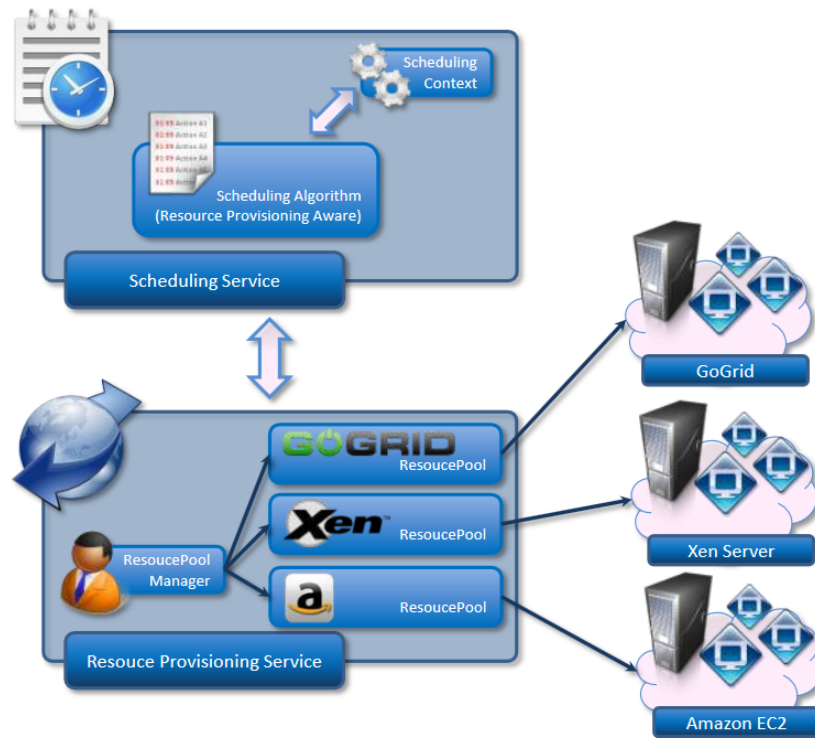


Abb. 5. Dynamische Bereitstellungsinfrastruktur [6]

verwaltet die Ausführung der Anwendungen. Der Scheduling Algorithmus ist dabei verantwortlich für die Planung der Ausführung der Anwendungen. Das bedeutet, dass er die in der Anwendung zu erledigenden Aufgaben dynamisch den verfügbaren Ressourcen zuteilt. In Aneka sind standardmäßig 2 Algorithmen verfügbar. Der *FixedQuereProvisioningAlgorithm* stellt ein Programmiermodell-weites Management zur Verfügung. Das bedeutet, dass die Menge an wartenden Anforderungen einen bestimmten Grenzwert nie überschreitet. Ebenso stellt er sicher, dass die Ressourcen bei Erreichen einer Untergrenze an wartenden Aufgaben wieder freigegeben werden. Dieser Algorithmus betrachtet dementsprechend alle vorhandenen Anwendungen und kontrolliert und optimiert die Performanz des gesamten Systems.

Der zweite vorhandene Algorithmus ist der *DeadlinePriorityProvisioningAlgorithm*. Er ist entworfen um dynamisch Ressourcen zuzuweisen, wenn die Qualitätsanforderungen einer bestimmten Anwendung nicht mehr erfüllt werden

können. Dieser Algorithmus betrachtet dementsprechend nur eine Anwendung und optimiert diese.

Der in der Abbildung dargestellte Scheduling Context stellt die Schnittstelle zwischen Scheduling Algorithmus und Service Implementierung dar. Er kommuniziert Scheduling-Entscheidung zum System und ermöglicht es dem Scheduling-Algorithmus dynamische Ressourcen anzufordern.

Der Resource Provisioning Service ist für die Bearbeitung von Ressourcenanfragen zuständig. Er kann als Black-Box angesehen werden, welche die Ressourcen bereitstellt, freigibt und den Ressourcenstatus sowie den Ressourcenpoolstatus überwacht.

3.3 Einordnung in die MAPE-K Architektur

Obwohl in der Beschreibung der dynamischen Ressourcenzuteilung in Aneka die MAPE-K Architektur nicht erwähnt wurde, ist diese dennoch indirekt erkennbar. Die Monitoring-Komponente ist in der Lastausgleich und Monitoring Komponente wiederzufinden, da diese die Systemlast überwacht und die Informationen an die Planungs und Management Facility weitergibt. Diese ist mit der Analyse-Komponente gleichzusetzen, da sie aus den erhaltenen Statusinformationen entscheidet, ob weitere Ressourcen benötigt werden oder nicht. Außerdem fällt die Budget Management Facility in die Analyse-Komponente, da die Informationen aus ihr mit in den Entscheidungsprozess einbezogen werden müssen. Die Planungs- und die Ausführungskomponente werden im konkreten Beispiel durch die dynamische Bereitstellungs-Factory übernommen und nicht getrennt behandelt. Das Wissen (Knowledge) aus der MAPE-K Architektur wird in Aneka durch die Service-Level-Agreements mit den Nutzern definiert und legt unter anderem fest, wann eine Anwendung weitere Ressourcen benötigt. Die Anzahl der benötigten Ressourcen ist an dieser Stelle auch in der Wissensdatenbank gespeichert und wird während der Laufzeit gelernt und angepasst (Selbstoptimierung). Die Zuordnung der einzelnen Aneka-Komponenten zu den Bestandteilen der MAPE-K Schleife ist in Abbildung 6 dargestellt.

4 Beispielablauf einer Optimierung

Um den Ablauf und die Ursachen einer ablaufenden Ressourcenoptimierung besser nachvollziehen zu können, wird im Folgendem ein Beispielablauf beschrieben. Die Optimierung findet dabei wie bereits in der Einleitung erwähnt primär dazu statt die Nutzerzufriedenheit zu steigern. Dies wird bewirkt durch eine hohe Performanz der Anwendung bei gleichzeitig minimalen Kosten. Die minimalen Kosten sind durch das Pay-as-you-go Konzept im Cloud Computing möglich, welches dafür steht, dass nur das gezahlt wird, was auch wirklich benutzt wird. Zweiter Punkt für die Optimierung ist der Energiebedarf und damit die Kosten für den Betreiber. Ausschlaggebend für den Start einer Optimierung sind in der Regel immer die vom Kunden mit dem Provider ausgehandelten SLAs. Wird eine Verletzung dieser festgestellt oder vorhergesagt, wird eine entsprechende

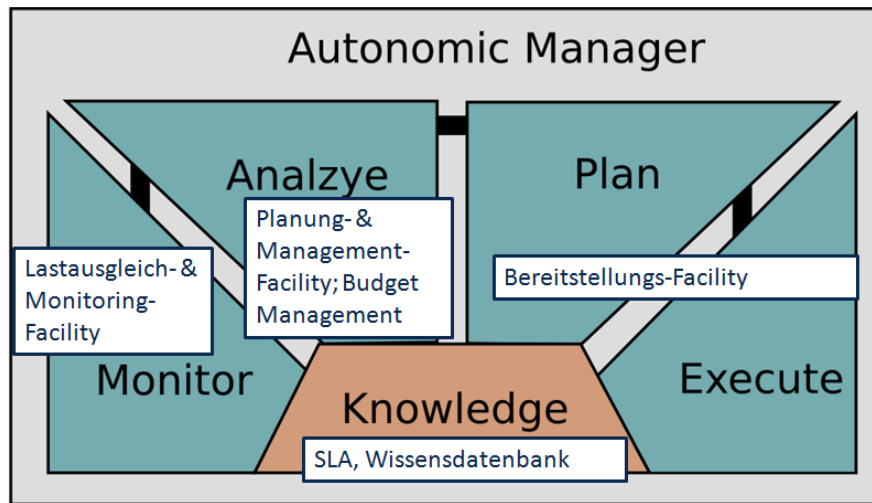


Abb. 6. Einordnung der Komponenten in MAPE-K

Optimierung angestoßen. Was dabei konkret getan werden muss, um die SLA Verletzung zu vermeiden und den gewünschten Zustand wieder herzustellen wird in der Wissenskomponente gespeichert. Diese Handlungen sind jedoch nicht direkt aus den SLAs ablesbar und es stellt nach wie vor ein Problem dar, wie SLAs auf Low-Level Anforderungen wie CPU und RAM abgebildet werden. Ein Beispiel für eine SLA Verletzung ist die Antwortzeit, welche in den SLAs als maximal 2 Sekunden definiert sein kann. Zur Lösung des Problems sei in diesem Beispiel angenommen, dass eine Erhöhung der Rechenleistung durch benutzen zusätzlicher CPUs die Antwortzeit verringert. Dementsprechend würden in diesem Fall eine in der Wissensquelle gespeicherte, gelernte Anzahl an CPUs automatisch zusätzlich der Anwendung zur Verfügung gestellt und die Antwortzeit entspricht wieder den SLAs. Um die Kosten und die Energie minimal zu halten wird für die Freigabe der zusätzlich allokierten Ressourcen neben der Antwortzeit auch die Auslastung der CPUs überwacht. Hierbei ist ebenso wie bei der Antwortzeit eine gelernte Grenze definiert ab welcher die CPUs wieder freigegeben werden um damit das zweite Optimierungskriterium "Kosten" zu minimieren. Anschließend ist der Normalzustand wieder hergestellt. Der beschriebene Ablauf wird auch durch Abbildung 7 visualisiert.

5 Fazit

Diese Arbeit hat gezeigt, dass die MAPE-K Referenzarchitektur nicht nur ein theoretisches Modell ist, sondern auch in der Praxis genutzt wird. Auch wenn in der Arbeit nur der Nutzen bei der automatischen Ressourcenzuweisung in der Cloud vorgestellt wurde lässt sich bereits erahnen, dass die MAPE-K Archi-

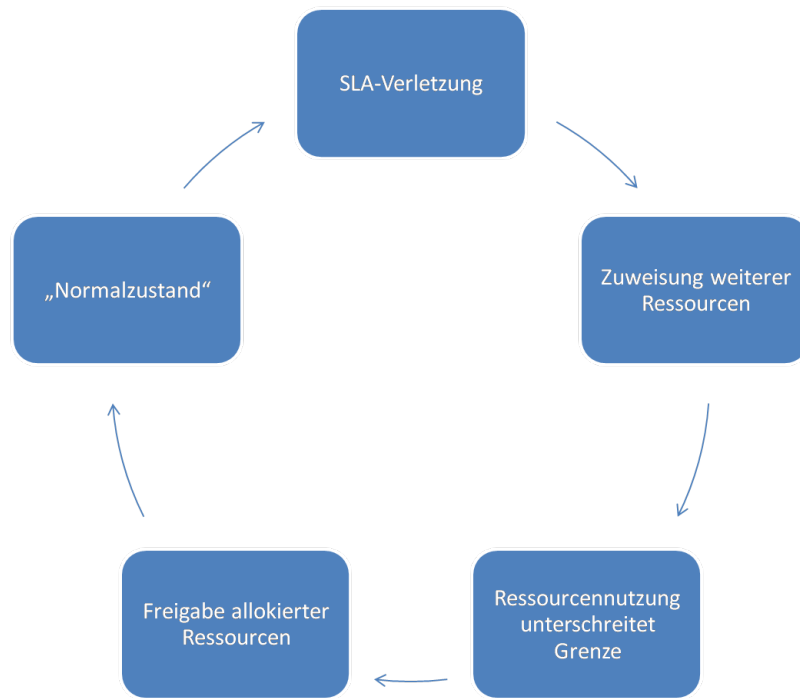


Abb. 7. Beispielhafter Ablauf einer Ressourcenzuweisung

tektur nicht nur dabei von großem Nutzen ist. Beispielhaft für weitere Anwendungen der Architektur seien autonomes Fahren oder Cyberphysical-Systems im Bereich der Industrie 4.0 zu nennen. Neben der Erkenntnis, dass die Architektur in der Praxis vielseitig einsetzbar ist soll auch das Bewusstsein gestärkt worden sein, dass die MAPE-K Architektur nur ein Referenzmodell darstellt und in der Praxis oftmals Anpassungen notwendig sind und getätigt werden. Im Beispiel Aneka welches näher vorgestellt wurde, wurden beispielsweise die Planungs- und Ausführungskomponente zusammengefasst. Weitere, bereits von IBM vorgestellte Erweiterungen, sind die Kombination und Orchestration mehrerer MAPE-K Schleifen in autonomen Managern [4].

Abschließend lässt sich festhalten, dass die MAPE-K Architektur in der Praxis bereits vielseitig genutzt wird, auch wenn dies nicht so benannt wird. Nichtsdestotrotz bieten sich in Zukunft mit großer Wahrscheinlichkeit weitere Anwendungs- und Anpassungsmöglichkeiten für diese Architektur, da die Automatisierung in allen Bereichen weiter voran schreitet.

6 Literatur

1. Peter M. Mell and Timothy Grance: **the NIST Definition of Cloud Computing**, NIST, Gaithersburg, MD, United States, 2011.
2. Qi Zhang, Cheng Lu and Raouf Boutaba, **Cloud computing: state-of-the-art and research challenges**, Journal of Internet Services and Applications, 2010.
3. Paul Horn, **Autonomic computing: IBM's Perspective on the State of Information Technology**, 2001.
4. IBM, **An Architectural Blueprint for Autonomic Computing**, IBM, 2005.
5. Manjrasoft Pty Ltd, **Aneka Architecure Documentation**. [Online]. Available: http://www.manjrasoft.com/aneka_architecture.html [Zugegriffen: 16. Mai 2017]
6. Manjrasoft Pty Ltd, **Aneka Dynamic Provisioning Documentation**. [Online]. Available: <http://www.manjrasoft.com/download/3.0/AnekaDynamicProvisioning.pdf> [Zugegriffen: 16. Mai 2017]